

# **Modelagem de Negócio: Uma Introdução e Prática**

**Vitor Hauck dos Santos**

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Orientador: Prof. Michel Heluey Fortuna



Juiz de Fora, MG  
Dezembro de 2009

# **Modelagem de Negócio: Uma Introdução e Prática**

**Vitor Hauck dos Santos**

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovada pela banca constituída pelos seguintes professores:

---

**Prof. Michel Heluey Fortuna** – orientador

Dr. em Engenharia de Sistemas e Computação, UFRJ, 2008

---

**Prof.<sup>a</sup> Alessandra Marta de Oliveira**

M. Sc. em Engenharia de Sistemas e Computação, UFRJ, 2003

---

**Prof.<sup>a</sup> Rita de Cássia Oliveira Estevam**

M. Sc. em Engenharia de Sistemas e Computação, UFRJ, 1990

Juiz de Fora, MG  
Dezembro de 2009

# Agradecimentos

A todos que colaboraram na realização deste estudo, incluindo os membros da empresa júnior de computação da UFJF - Base Três, que se mostraram dispostos a ajudar.

Ao meu orientador que esteve sempre presente na condução deste trabalho.

# Sumário

|   |            |
|---|------------|
| <b>Lista de Figuras .....</b>                                     | <b>v</b>   |
| <b>Lista de Tabelas.....</b>                                      | <b>vi</b>  |
| <b>Lista de Siglas e Abreviaturas.....</b>                        | <b>vii</b> |
| <b>Introdução.....</b>  | <b>9</b>   |
| <b>2. Modelagem de negócio.....</b>                               | <b>10</b>  |
| 2.1 Introdução .....  | 10         |
| 2.2 Modelagem de processos de negócio.....                        | 12         |
| 2.2.1 BPMN.....   | 13         |
| 2.2.2 Diagrama de Atividade da UML .....                          | 19         |
| 2.2.3 EPC.....  | 22         |
| 2.3 Modelagem de processo de negócio voltada para a execução..... | 24         |
| 2.3.1 BPEL .....  | 25         |
| 2.3.2 <i>WfMC Reference Model</i> .....                           | 28         |
| 2.3.3 YAWL .....  | 32         |
| 2.4 Modelagem de organização de negócio.....                      | 35         |
| 2.4.1 Relationship Maps .....                                     | 36         |
| 2.5 Modelagem de regras de negócio .....                          | 37         |
| 2.5.1 <i>RuleSpeak</i> .....                                      | 38         |
| 2.5.2 SBVR.....   | 39         |
| 2.6 Modelagem de motivação de negócio .....                       | 41         |
| 2.6.1 BMM .....   | 41         |
| 2.7 Considerações Finais .....                                    | 44         |
| <b>3. Modelagem de uma Empresa Real.....</b>                      | <b>46</b>  |
| 3.1 Introdução .....  | 46         |
| 3.2 Modelagem da Organização .....                                | 46         |
| 3.3 Modelagem das Regras de Negócio.....                          | 47         |
| 3.4 Modelagem das Motivações .....                                | 52         |
| 3.5 Modelagem dos Processos de Negócio.....                       | 53         |
| 3.5.1 Processo de desenvolvimento de um projeto .....             | 53         |
| 3.5.2 Processo <i>Trainee</i> .....                               | 54         |
| 3.5.3 Processo de Troca de Gestão.....                            | 54         |
| 3.6 Análise .....   | 52         |
| 3.7 Considerações Finais .....                                    | 53         |
| <b>4. Conclusão .....</b>   | <b>54</b>  |
| <b>Referências .....</b>  | <b>55</b>  |
| <b>Apêndice 1: Exemplo de uso de BPEL .....</b>                   | <b>58</b>  |
| <b>Apêndice 2: Exemplo de código XPD L .....</b>                  | <b>62</b>  |

## Lista de Figuras

|  |    |
|--|----|
| Figura 2.1 – Exemplo de BPD (OMG, 2008a) .....   | 18 |
| Figura 2.2 – Exemplo de utilização do Diagrama de Atividades (SAMPAIO, 2009) ...             | 22 |
| Figura 2.3 – Exemplo de utilização de EPC (WESKE, 2009) .....                                | 24 |
| Figura 2.4 – Exemplo de criação de um processo BPEL na plataforma Eclipse .....              | 28 |
| Figura 2.5 – Modelo de referência da WfMC (HAVEY, 2005).....                                 | 29 |
| Figura 2.6 – Diagrama de classes do XPDL (HAVEY, 2005).....                                  | 30 |
| Figura 2.7 – Exemplo de uso do programa JAWE (HAVEY, 2005) .....                             | 31 |
| Figura 2.8 – Exemplo de uso de YAWL (AALST e HOFSTEDE, 2003) .....                           | 34 |
| Figura 2.9 – Exemplo de uso de <i>Relationship Map</i> (BRIDGELAND e ZAHAVI, 2009),<br>..... | 37 |
| Figura 2.10 – Estrutura do BMM (OMG, 2008c) .....  | 42 |
| Figura 3.1 – <i>Relationship Map</i> da Base Três .....                                      | 47 |
| Figura 3.2 – BMM da gestão 2/2009 da Base Três .....   | 53 |
| Figura 3.3 – Primeira parte do BPD de desenvolvimento de projetos .....                      | 55 |
| Figura 3.4 – Segunda parte do BPD de desenvolvimento de projetos .....                       | 56 |
| Figura 3.5 - BPD do Processo Trainee .....   | 57 |
| Figura 3.6 - BPD do processo de eleições .....   | 58 |
| Figura A1.1 - Um processo de requisição de seguro .....                                      | 60 |

## Lista de Tabelas

|   |    |
|---|----|
| Tabela 2.1 – Elementos do BPMN .....                    | 14 |
| Tabela 2.2 – Elementos do diagrama de atividades.....   | 19 |
| Tabela 2.3 – Elementos do EPC .....                     | 23 |
| Tabela 2.4 – Elementos do BPEL.....                     | 26 |
| Tabela 2.5 – Elementos do XPDL.....                     | 30 |
| Tabela 2.6 – Elementos do YAWL .....                    | 32 |
| Tabela 2.7 – Elementos do <i>Relationship Map</i> ..... | 36 |
| Tabela 2.8 – <i>RuleSpeak</i> .....                     | 38 |
| Tabela 2.9 – Palavras chaves do SBVR (OMG, 2008b) ..... | 40 |
| Tabela 2.10 – Exemplo de uso do BMM.....                | 44 |
| Tabela 3.1 – Tradução das palavras chaves do SBVR ..... | 48 |
| Tabela 3.2 – Regras da Base Três.....                   | 48 |

## Lista de Siglas e Abreviaturas

- BEA:** *Bureau of Economic Analysis*
- BMM:** *Business Motivation Model*
- BPD:** *Business Process Diagram*
- BPEL:** *Business Process Execution Language*
- BPM:** *Business Process Management*
- BPMI:** *Business Process Management Initiative*
- BPMN:** *Business Process Model Notation*
- EPC:** *Event-driven process chain*
- OASIS:** *Organization for the Advancement of Structured Information Standards*
- OMG:** *Object Management Group*
- PD4J:** *Process Definition for Java*
- SBVR:** *Semantics of Business Vocabulary and Business Rules*
- UML:** *Unified Modeling Language*
- WAPI:** *Workflow Application Programming Interface*
- WFMC:** *Workflow Management Coalition*
- WfXML:** *Workflow Xml*
- XML:** *eXtensible Markup Language*
- XPDL:** *XML Process Definition Language*
- YAWL:** *Yet Another Workflow Language*

## **Resumo**

Este trabalho apresenta uma visão geral da modelagem de negócio, um tema relativamente novo, mas que tem crescido em importância recentemente. Por ser novo, o tema ainda está em desenvolvimento, muitos padrões são recentes e existem aspectos de modelagem que não possuem padrões, sendo apoiados apenas por iniciativas individuais. Além de apresentar a modelagem de negócio e justificar sua importância, o trabalho descreve algumas das linguagens usadas para a construção de modelos de negócio, de forma breve e com exemplos, permitindo ao leitor ter conhecimento do que existe atualmente nessa área. Complementando esse estudo, é feita uma modelagem parcial da empresa júnior de informática da UFJF - Base Três.

## **Palavras-Chave**

Modelagem de Negócio, Modelo de Processo de Negócio, BPMN, Gerência de Processo de Negócio.

## Introdução

Modelagem de negócio é um tema novo, e que ainda está em desenvolvimento. Surgiu com a recente necessidade das organizações de lidarem com a complexidade cada vez maior dos seus negócios. O ambiente econômico atual é marcado por grandes corporações, onde a competição pelo mercado é acirrada, o que obriga que cada uma delas esteja sempre em constante evolução para se manter competitiva. São comuns as aquisições entre organizações, o que obriga um processo de integração entre elas. Todo esse dinamismo é difícil de ser gerenciado, e modelos de negócio apóiam essa gerência.

Inicialmente existiram iniciativas individuais, com algumas organizações criando linguagens para modelar o seu próprio negócio ou para vender ferramentas de apoio à utilização dessas linguagens por outras organizações. Atualmente está acontecendo a evolução dessas linguagens proprietárias para padrões, exatamente como aconteceu com a modelagem de software na década de 1990.

O fato do tema ser recente e o interesse nele estar crescendo foi o que motivou esse estudo. Quando se pesquisa por modelagem de negócio, o que se encontra é uma grande variedade de siglas, representando linguagens, ferramentas e até organizações responsáveis. Essa variedade revela que o tema ainda tem um longo caminho a seguir até que se chegue a um consenso sobre qual linguagem usar para modelar um determinado aspecto de uma organização.

A proposta deste trabalho é fornecer ao leitor um panorama do que é hoje utilizado para a modelagem de negócio, permitindo que o mesmo conheça algumas linguagens através de descrições e exemplos.

O trabalho foi dividido em quatro capítulos, sendo o primeiro deles essa introdução. O Capítulo 2 apresenta várias linguagens para a modelagem de diferentes aspectos de uma organização; cada seção dele descreve e exemplifica o uso de uma ou mais linguagens utilizadas na modelagem de um determinado aspecto do negócio ou organização. O Capítulo 3 busca consolidar o que foi visto nos dois anteriores, através da modelagem do negócio de uma empresa real, a empresa júnior de informática da UFJF – Base Três, apresentando também uma análise das principais dificuldades encontradas na construção dos modelos. Por fim, o Capítulo 4 conclui este trabalho, retomando o que foi abordado, destacando suas limitações e sugerindo trabalhos futuros.

## **2. Modelagem de negócio**

### **2.1 Introdução**

Um modelo é uma representação simplificada de uma realidade complexa que possui um propósito. Modelos são muito utilizados no dia-a-dia das pessoas, como por exemplo: mapas, manuais de montagem, guias de programação de TV etc. Na engenharia os modelos são fundamentais. Cada ponte, carro, avião ou circuito integrado foi construído usando um modelo. O propósito de um modelo de engenharia é facilitar a comunicação com os clientes, mostrar como o produto vai ser e possibilitar o trabalho em conjunto de pessoas com diferentes responsabilidades no projeto.

A engenharia de software é uma disciplina nova, e apenas recentemente modelos de software têm sido construídos. No final da década de 1980 várias linguagens de modelagem foram criadas e, nos anos 90, a demanda por modelos aumentou, ocasionando a criação da UML, uma tentativa de unificar as linguagens de modelagem existentes até então. Com a adoção da UML como um padrão pela OMG em 1997, a criação de modelos que utilizam essa linguagem se tornou uma prática comum para os engenheiros de software atuais.

Historicamente, o uso de modelos de negócio foi bastante limitado. Quadros organizacionais e demonstrações contábeis são utilizados desde a antiguidade mas, tirando essas duas exceções, modelos de negócio raramente eram utilizados. Até recentemente, a comunicação entre as pessoas do negócio era feita utilizando palavras ou planilhas. Porém, nos últimos quinze anos, um número cada vez maior de pessoas tem criado modelos de negócio: modelos de processos, da organização, dos objetivos e estratégias e das políticas e regras.

É visível uma progressão de modelos e ferramentas proprietárias para padrões, a mesma progressão que aconteceu com a engenharia de software na década de 90. BRIDGELAND e ZAHAVI (2009) esperam que, em dez anos, a modelagem de negócio seja uma prática tão comum para as pessoas de negócio quanto a modelagem de software é hoje para os engenheiros de software.

A criação de modelos é uma atividade trabalhosa e que demanda tempo e dinheiro. Por isso devem existir boas justificativas para realizá-la. Ainda segundo

BRIDGELAND e ZAHAVI (2009), modelos de negócio dão suporte a oito importantes atividades não mutuamente exclusivas. São elas:

1. Comunicação entre pessoas: Um negócio é uma atividade de comunicação intensa. Hoje em dia, a maioria das pessoas do negócio utiliza palavras, apresentações ou desenhos improvisados como forma de comunicação. Essas soluções funcionam, mas possuem deficiências. Palavras podem conter ambigüidade e apresentações podem ser interpretadas de diferentes maneiras pelas pessoas. Diferentemente, um modelo é construído com base em uma linguagem que estabelece rigorosamente quais são seus elementos e o que cada um deles significa, eliminando assim a ambigüidade;
2. Treinamento e aprendizado: Aprender com a experiência dos outros é mais rápido, barato e menos arriscado que aprender por tentativa e erro. Portanto, é a forma utilizada para preparar um novo funcionário. O treinamento é muito facilitado se a organização possuir modelos, pois um modelo de processo de negócio, por exemplo, descreve uma atividade passo a passo e pode ser usado para instruir um novato;
3. Persuasão e vendas: Persuadir é usar a comunicação a serviço de um propósito. Alguém que construiu um modelo demonstra que possui um conhecimento aprofundado de um determinado tema. Os clientes normalmente têm problemas, e conhecer bem esses problemas é fundamental para se ter credibilidade. Esse conhecimento também leva a melhores soluções;
4. Análise da situação do negócio: Modelos simplificam a complexa realidade de um ambiente organizacional e, por isso, possibilitam uma visão melhor do negócio. Isso é fundamental para se tomar boas decisões.
5. Gerência de conformidade: Um negócio tem que estar de acordo com as leis em vigor. Também tem que cumprir contratos, e funcionários tem que trabalhar de acordo com a política da empresa. Esse tipo de gerência pode ser facilitado se existirem modelos, por exemplo, que descrevam como cada funcionário deve agir para manter a conformidade;
6. Desenvolvimento de requisitos de software: requisitos são fundamentais para o desenvolvimento de software e se não estiverem bem descritos e detalhados o projeto pode falhar. Porém, os usuários finais nem sempre sabem descrever com os detalhes necessários os requisitos, apesar de conhecerem bem a forma como realizam o seu trabalho. Modelos de negócio capturam os detalhes necessários

de uma maneira que é bem compreendida tanto pelos usuários finais quanto pelos desenvolvedores.

7. Execução direta em *engines* de software: Além de facilitar a tomada de decisão pelas pessoas, modelos podem ser usados por softwares capazes de automatizar a tomada de decisões. A vantagem disso é que caso alguma regra mude, não é necessário modificar o software, apenas o modelo;
8. Gerência de conhecimento e reuso: Gerência do conhecimento é a prática sistemática de capturar o conhecimento de certas pessoas do negócio, de forma que esse conhecimento possa ser usado por outras. Os detalhes das regras, atividades de um processo, objetivos e táticas de um negócio são todos capturados e explicitados na criação de um modelo de negócio.

Como já foi dito, dois tipos de modelos de negócio têm sido usados por muito tempo: o organograma e as demonstrações contábeis. Os organogramas descrevem hierarquias, e foram primeiramente utilizados no antigo Egito, China e Pérsia para representar exércitos e burocratas do governo. Demonstrações contábeis existem desde o período da renascença, com o surgimento da contabilidade moderna. É um conjunto de informações que refletem a situação financeira da empresa. Esses dois modelos clássicos são úteis, mas não refletem todos os aspectos da organização.

Recentemente, novos modelos têm se tornado importantes. Eles visam complementar os modelos clássicos, focando em aspectos diferentes da complexa realidade de negócio. São eles: modelos de processos de negócio, modelos de organização de negócio, modelos de regras de negócio e modelos de motivação de negócio. As seções seguintes procuram caracterizar e descrever cada um desses tipos de modelos.

## **2.2 Modelagem de processos de negócio**

Processo é qualquer atividade ou grupo de atividades que recebe uma entrada, adiciona valor a ela, e provê uma saída ao “consumidor” interno ou externo. Processo de negócio é um grupo de atividades logicamente relacionadas que usam os recursos da organização para prover resultados em apoio aos objetivos da organização (HARRINGTON, 1991).

Segundo JOHANSSON (1993), um processo de negócio é um conjunto de atividades relacionadas ao negócio que recebe uma ou mais entradas e as transforma

para criar uma saída. Idealmente, a transformação que ocorre no processo deve adicionar valor à entrada e criar uma saída que é mais útil e efetiva para o beneficiário.

Modelagem de processos de negócio é uma abstração de processos de negócio representada tipicamente em forma gráfica (ACTANO, 2009).

Um modelo de processo de negócio dá suporte a todas as oito atividades já mencionadas na Seção 2.1.

A seguir, são descritas algumas notações para a modelagem de processos de negócio.

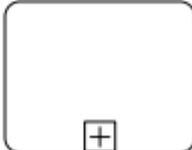
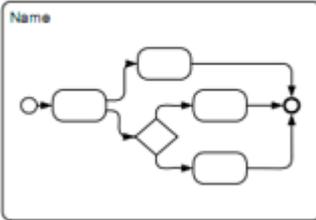
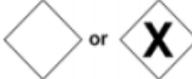
### **2.2.1 BPMN**

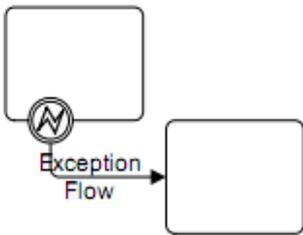
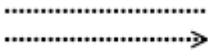
O BPMN (*Business Process Modeling Notation*) foi publicado em 2004 pela BPMI (Business Process Management Initiative) (BPMI, 2009), organização sem fins lucrativos que objetiva promover e desenvolver a gerência de processo de negócio com o estabelecimento de padrões. O BPMN tornou-se um padrão mantido pela OMG em 2006 e atualizado em 2008 (OMG, 2008a). O objetivo primário do BPMN é prover uma notação que seja compreensível por todos os usuários do negócio, desde os analistas que criam os primeiros rascunhos dos processos, passando pelos técnicos responsáveis por implementar a tecnologia que vai realizar esses processos, até as pessoas do negócio responsáveis por administrar e monitorar os processos. Portanto BPMN cria uma ponte padronizada para a lacuna que existe entre o design do processo de negócio e implementação do mesmo.

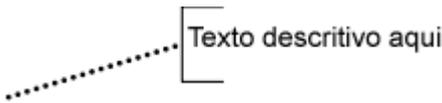
BPMN utiliza um tipo de diagrama para modelagem de processos de negócio denominado BPD (*Business Process Diagram*). As categorias de elementos de um BPD são: objetos de fluxo, objetos de conexão, raias e artefatos. Objetos de fluxo são os principais elementos gráficos para definir o comportamento de um processo de negócio. São eles: eventos, atividades e *gateways*. Objetos de conexão são usados para fazer ligações entre objetos de fluxo ou entre objetos de fluxo e artefatos. Podem ser de três tipos: fluxo de seqüência, fluxo de mensagem, ou associação. Raias servem para agrupar elementos primários de modelagem. Podem ser *pools* ou *lanes*. Por fim, artefatos são usados para prover informação adicional sobre o processo e se subdividem em objeto de dados, grupo, ou anotação (mas modeladores são livres para adicionar outros artefatos, se necessário).

A Tabela 2.1 trás uma visão geral dos elementos que compõem um BPD, descrevendo-os.

Tabela 2.1 – Elementos do BPMN (adaptado de OMG (2006))

| Elemento  | Descrição   | Notação   |
|-----------|---|---|
| Evento    | Um evento é algo que acontece durante o curso de um processo de negócio. Eventos afetam o fluxo de um processo e usualmente possuem uma causa (ativador) ou um impacto (resultado). Podem ser classificados de acordo com a forma que afetam o fluxo: iniciais, intermediários, finais. Podem apresentar um ícone, representando o tipo de evento que está ocorrendo. | <p>Inicial </p> <p>Intermediário </p> <p>Final </p>  |
| Atividade | Atividade é um termo genérico para o trabalho que uma companhia realiza. Uma atividade pode ser atômica ou composta. Os tipos de atividades são: processos, subprocessos e tarefas. Tarefas e subprocessos são representados por retângulos de bordas arredondadas e processos ficam contidos em um <i>pool</i>   | <p>Tarefa (atômica) </p> <p>Processo comprimido </p> <p>Processo Expandido </p>   |
| Gateway   | Um gateway é usado para controlar a convergência e a divergência de múltiplos fluxos. Ícones dentro do losango indicam o comportamento do controle de fluxo.  | <p>Exclusivo baseado em dado  or </p> <p>Exclusivo baseado em evento </p> <p>Inclusivo </p> <p>Complexo </p> <p>Paralelo </p> |

| Elemento           | Descrição   | Notação   |
|--------------------|---|---|
| Fluxo de Sequência | Um fluxo de seqüência é usado para mostrar a ordem que as atividades são efetuadas em um processo. Fluxo não controlado é o caso mais simples, quando não é afetado por condições nem atravessa <i>gateways</i> . Fluxo condicional possui expressões condicionais que avaliam se o fluxo vai ou não ser tomado. Fluxo padrão vai ser tomado sempre que nenhum fluxo condicional for verdadeiro. Fluxo de exceção é baseado na ocorrência de um evento intermediário. | <p>Não-controlado </p> <p>Condicional </p> <p>Padrão </p> <p>Exceção </p> |
| Fluxo de mensagem  | É usado para mostrar o fluxo de mensagem entre duas entidades que estão preparadas para recebê-las e enviá-las. Em BPMN dois <i>Pools</i> separados em um diagrama representam duas entidades.  |    |
| Associação         | É usado para associar informação com objetos de fluxo. Texto e objetos gráficos não pertencentes ao fluxo podem ser associados dessa maneira.   |   |
| <i>Pool</i>        | Um <i>Pool</i> representa um participante do processo. Também funciona como raia, um container gráfico que separa um conjunto de atividades.  |   |
| <i>Lane</i>        | Subdivisão em um pool, com o propósito de organizar as atividades.  |   |

| Elemento        | Descrição   | Notação   |
|-----------------|---|---|
| Objeto de dados | São considerados artefatos, pois não tem efeito direto sobre o fluxo de seqüência ou mensagem do processo, mas provêm informações sobre quais atividades devem ser efetuadas e o que elas produzem. |  |
| Grupo           | Serve para agrupar atividades de uma mesma categoria. Não afeta o fluxo de seqüência das atividades do grupo. Grupos podem ser usadas para análise de documentação.                                 |   |
| Anotação        | Um mecanismo para o modelador prover informação adicional ao leitor do diagrama.  |  |

A seguir, um BPD é apresentado como exemplo (Fig. 2.1). Nele é possível visualizar a utilização de várias das estruturas descritas na Tabela 2.1 para descrever um processo de votação por e-mail, que existe em um grupo de trabalho. O processo é descrito do ponto de vista do administrador da lista de problemas, de forma que os votantes são participantes externos que se comunicarão com as atividades usando fluxos de mensagens.

O processo começa na sexta-feira, por isso o ícone de relógio no evento inicial, representando tempo. O administrador recebe a lista de problemas e analisa se algum deles já está pronto para ser votado. Caso nenhum esteja, o fluxo padrão é seguido levando a um evento final. Caso haja problemas prontos, o fluxo segue para a atividade *Ciclo de discussões*, que está representada de forma comprimida, sendo descrita em mais detalhes em outro diagrama (não incluído nesse trabalho). Essa atividade dispara duas mensagens para os votantes, informando quais problemas devem ser votados e quando o prazo da votação acaba. Em seguida existe uma atividade que anuncia os problemas e envia uma mensagem aos votantes. Segue-se outra atividade comprimida, coletar votos. Essa atividade troca mensagem com os votantes nos dois sentidos, eles

votam e ela anuncia o prazo para o fim da votação. Existe um fluxo de exceção por tempo nessa atividade, depois de uma semana o prazo acaba seguindo-se a atividade de preparar os resultados. Dessa atividade saem dois fluxos paralelos, que são a postagem dos resultados no *website* e o envio dos resultados por e-mail aos votantes. Como o fluxo é paralelo essas atividades acontecem concomitantemente. Um *gateway* é usado para convergir os dois fluxos, seguido por outro, dessa vez uma tomada de decisão. Se participantes suficientes votaram e não há problemas sem maioria o fluxo padrão é seguido levando a um evento final. Caso não haja votantes suficientes e os membros não foram alertados, outro fluxo padrão é tomado para a atividade que anuncia novamente a necessidade de votar e em seguida direciona o fluxo para a atividade coletar votos. Caso os membros já tenham sido alertados, o fluxo segue para a atividade que reduz o número de votantes e recalcula os resultados. Essa atividade é ligada no *gateway* que avalia a existência de problemas sem maioria. Se ainda existirem problemas sem maioria, um novo *gateway* avalia se é a segunda vez que isso ocorre. Se for, o fluxo é direcionado para a atividade do ciclo de discussão novamente, senão, o fluxo padrão é tomado, que leva para uma atividade que enquadra outras duas, uma para reduzir o problema a duas soluções e outra para enviar e-mail aos participantes que tiverem que mudar seus votos. Daí o fluxo segue para a atividade de anúncio dos problemas.

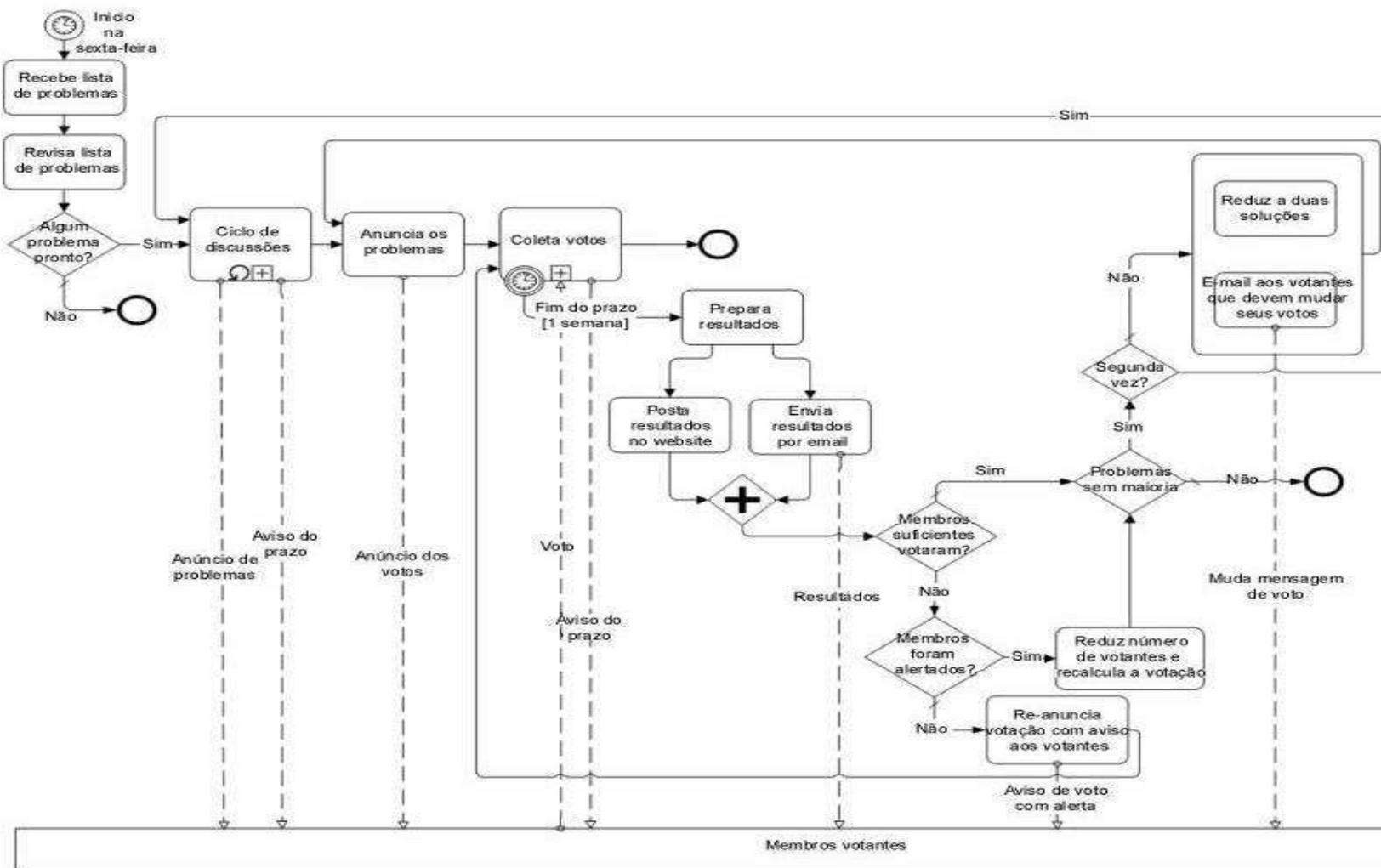


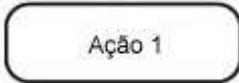
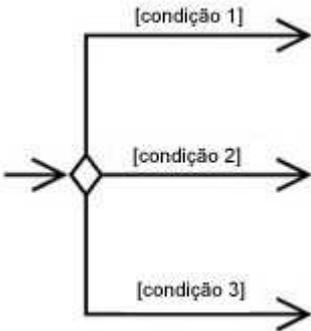
Figura 2.1 – Exemplo de BPD (OMG, 2008a)

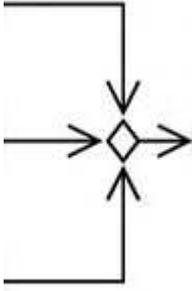
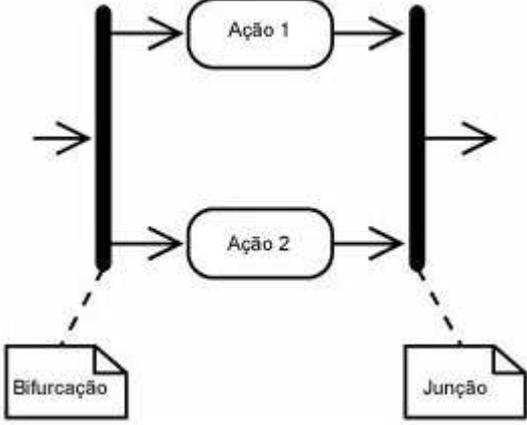
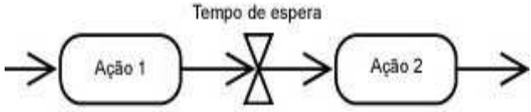
## 2.2.2 Diagrama de Atividade da UML

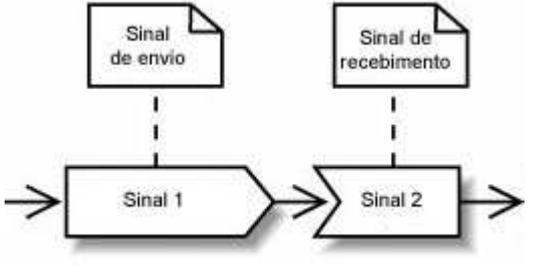
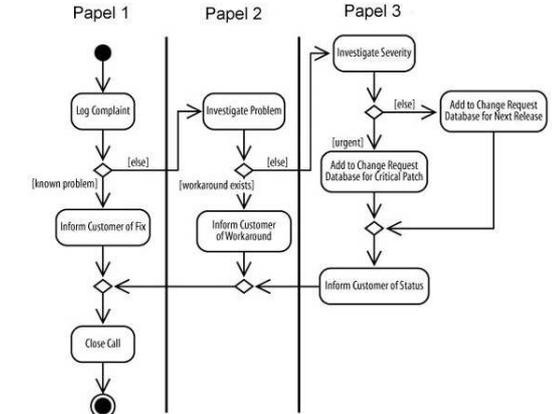
A UML (*Unified Modeling Language*) teve sua primeira especificação publicada pela OMG em 1997, em uma tentativa de criar um padrão para linguagens de modelagem de software, e teve colaboração de grandes nomes dessa área, como Jacobson, Booch e Rumbaugh. Atualmente, a UML é de fato um padrão para modelagem de software e sua especificação define vários diagramas, cada um deles permitindo modelar diferentes aspectos do sistema. Dentre eles, o Diagrama de Atividade que permite ao modelador especificar uma atividade que contribui para o alcance dos objetivos dos usuários de um sistema. Portanto, diagramas de atividade se prestam à modelagem de processos de negócio. Existem, inclusive, ferramentas capazes de executar processos de negócio descritos através de diagramas de atividade (HAMILTON e MILES, 2006).

A tabela 2.2 descreve a notação e o significado dos elementos de um diagrama de atividades.

Tabela 2.2 – Elementos do diagrama de atividades

| Elemento | Descrição  | Notação  |
|----------|--|--|
| Ação     | Um passo realizado em direção a conclusão de um processo. Pode ser um cálculo ou uma tarefa, por exemplo.  |  |
| Decisão  | É usada quando se deseja executar uma seqüência de ações diferentes dependendo de condições. É simbolizada como um losango com uma aresta de entrada e múltiplas arestas de saída. Cada aresta de saída contém uma condição de guarda, escrita entre colchetes, que é uma sentença que ao ser avaliada retorna verdadeiro ou falso. O caminho seguido é aquele cuja condição de guarda retorna verdadeiro. |  |

|                            |  |   |
|----------------------------|--|---|
| <p>União</p>               | <p>É representada por um losango, como a decisão. Apresentam-se vários fluxos de entrada e apenas um fluxo de saída. Serve para unir os fluxos que chegam ao losango.</p>  |     |
| <p>Bifurcação e junção</p> | <p>São usados quando se deseja representar ações que ocorrem paralelamente. Ambos são representados por uma barra grossa vertical e podem ser diferenciados pelo número de fluxos de entrada e de saída. A bifurcação possui apenas um fluxo de entrada e vários de saída, e a junção possui vários fluxos de entrada e apenas um de saída. Uma junção somente acontece quando todos os fluxos de entrada são finalizados.</p> |     |
| <p>Eventos de tempo</p>    | <p>São representados por uma ampulheta e utilizados para representar um tempo de espera entre ações, ou ações que ocorrem a cada intervalo de tempo. Quando possuem fluxo de entrada, indicam tempo de espera; caso contrário, indicam ações que ocorrem periodicamente.</p>   |   |
| <p>Objetos</p>             | <p>São usados quando objetos de dados são um aspecto importante do processo modelado. Representam um objeto que está disponível em um ponto particular de um processo e pode ser usado, criado ou modificado pelas ações ao seu redor. É representado como um retângulo.</p>   |  |

|                                  |   |  |
|----------------------------------|---|--|
| <p><b>Sinais</b></p>             | <p>Representam interações com participantes externos. Sinais são mensagens que podem ser recebidas ou enviadas. Um sinal de recebimento aguarda pela mensagem de um agente externo para prosseguir o fluxo.</p> |  |
| <p><b>Partições ou raias</b></p> | <p>Mostram quais participantes são responsáveis por quais ações. Partições dividem o diagrama em linhas ou colunas que contém as ações que são realizadas por um participante.</p>                              |  |

O diagrama de atividade da Figura 2.2 modela o processo de retirada de dinheiro em um caixa eletrônico e exemplifica o uso de várias das estruturas descritas na Tabela 2.2.

O diagrama é dividido em três partições, que representam os participantes do sistema: o cliente, o caixa eletrônico e o banco. O processo começa no cliente, quando ele insere o cartão no caixa eletrônico e, em seguida, digita a senha. O banco então verifica a senha digitada, com o uso de uma decisão com duas condições de guarda: uma para senha correta, que direciona o fluxo para a ação “Digita quantia”; outra para senha incorreta, que direciona o fluxo para uma união anterior à ação “Ejeta cartão”. O cliente digita a quantia que deseja retirar e o banco verifica o saldo desse cliente, o que leva a outra decisão com duas condições de guarda. Caso o saldo seja maior ou igual à quantia digitada, o fluxo segue para uma bifurcação. Caso contrário, segue para uma união imediatamente anterior à ação “Mostra saldo”. A bifurcação expressa a idéia de paralelismo e, portanto, as ações “Debita conta” e “Retira dinheiro” acontecem concomitantemente. Os fluxos de ambas seguem para uma junção. Isso significa que o fluxo somente vai seguir em frente quando ambas as ações forem concluídas. Então o caixa eletrônico mostra o saldo para o cliente e ejeta o cartão. O cliente retira o cartão do caixa e a atividade termina.

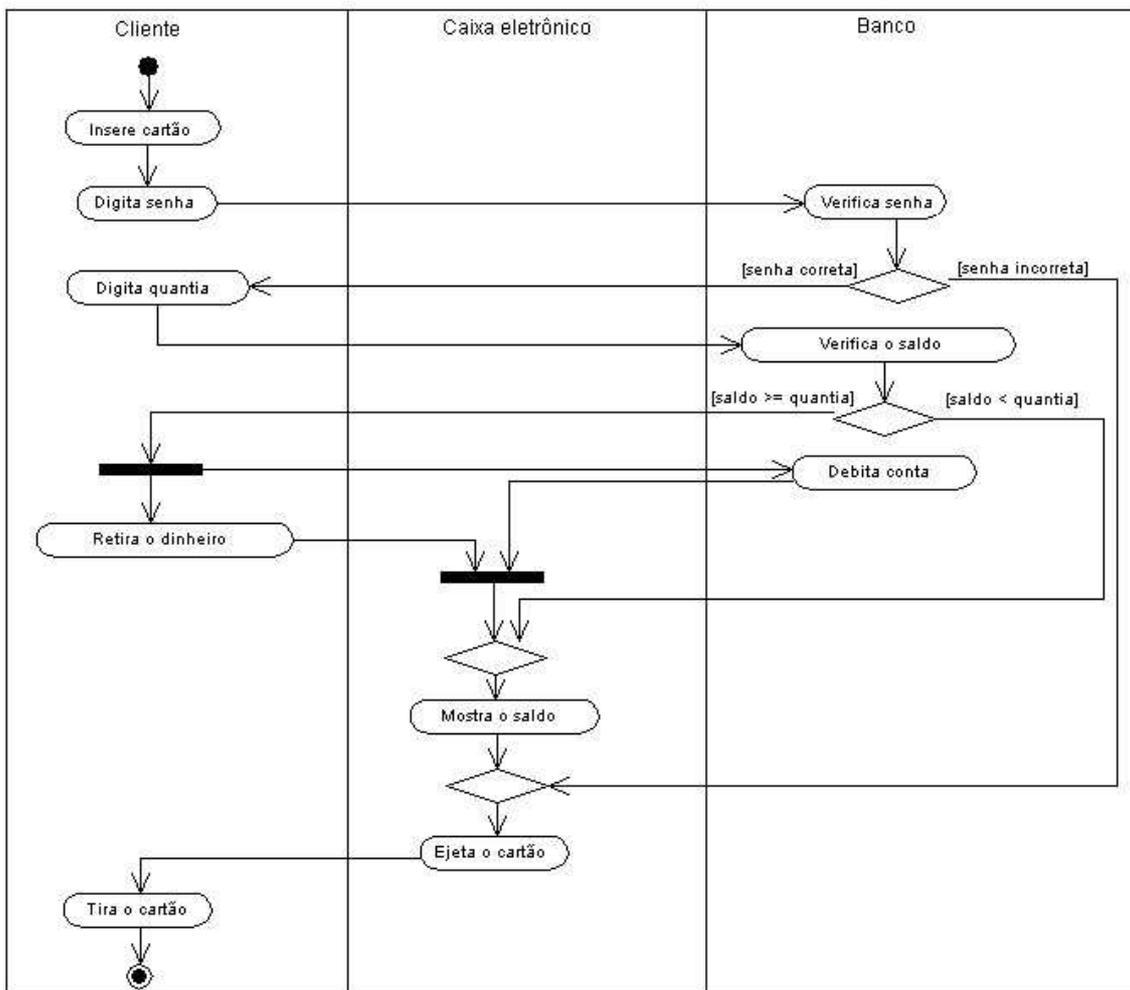


Figura 2.2 – Exemplo de utilização do Diagrama de Atividades (SAMPAIO, 2009)

### 2.2.3 EPC

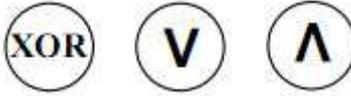
A modelagem denominada de *Event-driven Process Chain* (EPC)<sup>1</sup> foi desenvolvida pelo *Institute for Information Systems* (IWi) da Universidade de Saarland, na Alemanha. É uma linguagem gráfica muito utilizada em ferramentas e *frameworks* relacionados a processos de negócio, e teve bastante influência como notação de modelagem nos anos 90. Trata-se de uma cadeia de eventos e funções que suporta execução de processos paralelos. Um recurso notável do EPC é a presença de operadores lógicos. (KO *et al*, 2009).

<sup>1</sup> Uma tradução seria: “Cadeia de processos guiados por eventos”.

EPC é baseado nos conceitos de redes estocásticas e redes de Petri, e procura ser fácil de entender e usar por pessoas de negócio. (KORHERR e LIST, 2006).

A Tabela 2.3 apresenta a notação e uma breve descrição dos elementos da EPC.

Tabela 2.3 – Elementos do EPC

| Elemento                        | Descrição  | Notação   |
|---------------------------------|--|---|
| Função                          | Bloco básico de construção do diagrama. Corresponde a uma tarefa (passo de um processo) que deve ser executada.                                    |    |
| Evento                          | Descrevem a situação antes e depois que a função é executada. Funções são conectadas por eventos. Um evento pode agir como condição de uma função. |    |
| Operador lógico                 | Conecta funções e eventos. Pode ser de três tipos. OR (ou lógico), AND (e lógico), XOR (ou exclusivo).   |     |
| Unidade organizacional ou papel | O responsável por determinada função   |   |
| Objetos de informação           | Representam dados de entrada que servem de insumo para uma função, ou dados de saída produzidos por uma função.                                    |  |

O diagrama presente na Figura 2.3 ilustra o uso dos elementos descritos na Tabela 2.3, através da modelagem de um processo de chegada e verificação de produtos em uma empresa.

O processo começa com o evento que indica a chegada dos produtos. A unidade organizacional “Depto. de chegada de produtos” realiza a função “Verifica os produtos”, e o fluxo segue para um operador lógico XOR. A presença desse operador indica que apenas um dos caminhos vai ser tomado: os produtos estão OK ou não. Caso estejam, o “Depto. de chegada de produtos” “Armazena os produtos recebidos” e o processo acaba. Caso contrário, o fluxo segue para um operador lógico do tipo AND. Esse operador indica que os três caminhos são tomados: o “Depto. de gerência de qualidade” realiza a função “Cria relatório de inspeção” gerando um objeto de informação, o “Relatório de inspeção”; o “Depto. de problemas c/ produtos” realiza a função “Devolve produtos” e o “Depto. de reclamações”, a função “Envia pedido de

substituição”, gerando um objeto de informação denominado “Pedido de substituição”. Os eventos após cada uma dessas três funções indicam a situação quando elas terminam

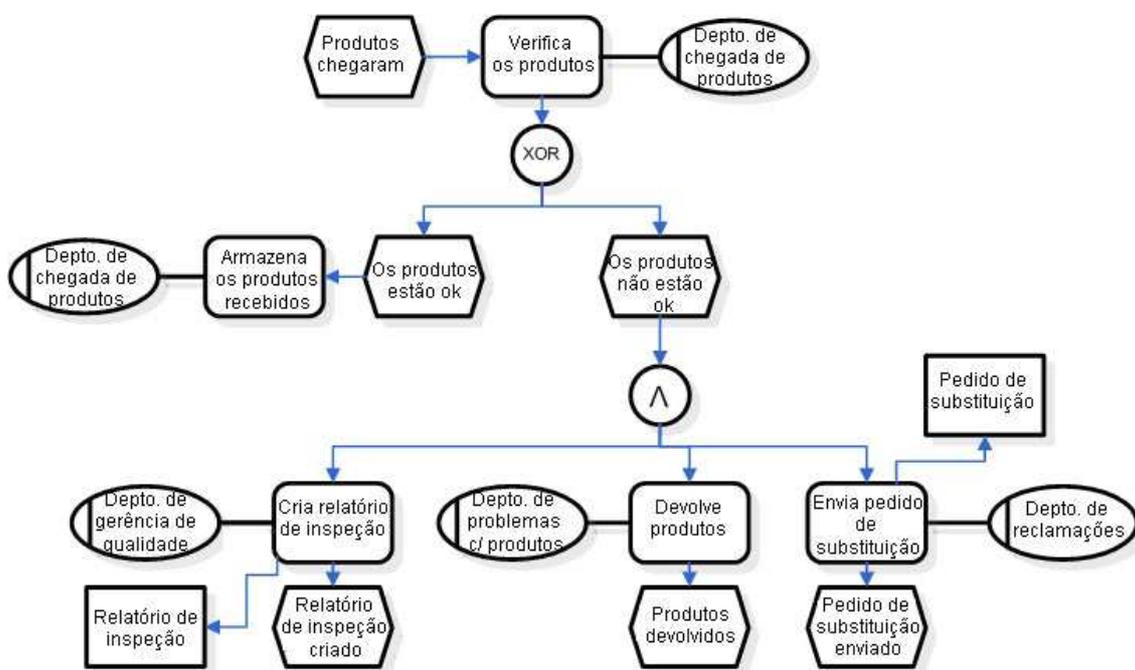


Figura 2.3 – Exemplo de utilização de EPC (WESKE, 2009)

### 2.3 Modelagem de processos de negócio voltada para a execução

As linguagens descritas na Seção 2.2 permitem construir diagramas, que são úteis em vários aspectos, principalmente na documentação dos processos e na comunicação entre pessoas. Muitas vezes, no entanto, deseja-se automatizar esses processos.

É sabido que processos de negócio têm uma natureza dinâmica, precisando ser continuamente adaptados e melhorados para que a organização possa se manter competitiva. Isso justifica a utilização de uma linguagem de execução de processos, capaz de permitir a definição do processo e sua posterior execução. Assim, qualquer mudança que aconteça em um processo de negócio pode ser rapidamente implementada alterando-se a sua definição executável, poupando tempo e garantindo o comprometimento de todos os envolvidos com a mudança.

Além de possibilitar a automatização dos processos de negócio, uma linguagem de execução de processos deve permitir uma forma padronizada de comunicação entre

as aplicações que compõem os processos, e prover integração suficiente para que novas aplicações possam ser facilmente incorporadas a eles.

Essa seção abordará algumas linguagens disponíveis para execução de processos de negócio. Além das três linguagens abordadas (BPEL, XPDL e YAWL), vale destacar também BPML (*Business Process Modeling Language*), desenvolvida pela BPMI (também responsável pelo BPMN), uma linguagem inicialmente promissora mas que atualmente está obsoleta devido ao maior apoio da indústria ao BPEL (KO *et al*, 2009).

### 2.3.1 BPEL

*Business Process Execution Language for Web Services* (BPEL4WS ou WS-BPEL, normalmente abreviada para BPEL) (JURIC *et al.*, 2006) é uma linguagem para definição e execução de processos de negócio através de *Web Services*. Foi escrita pela IBM, Microsoft e BEA e subseqüentemente padronizada pela OASIS (OASIS, 2009). As raízes conceituais do BPEL coincidem com as iniciativas anteriores na área de BPM dessas três companhias: WSFL (baseada em redes de Petri) da IBM (LEYMANN, 2001), XLANG (baseada em Calculo Pi) da Microsoft (MICROSOFT, 2009) e PD4J da BEA (SUN, 2009). Conseqüentemente, BPEL tira proveito dessas duas teorias.

BPEL não possui uma notação gráfica, representa processos de negócio através de uma linguagem baseada em XML. É uma extensão dos *Web Services* tradicionais, permitindo orquestração e coreografia (MATHEW e SARANG, 2006) e possibilitando construir processos de negócio que interagem entre si através de *Web Services*. Um processo descrito em BPEL é constituído de dois tipos de arquivos: arquivos de extensão WSDL (*Web Services Definition Language*) e arquivos com extensão BPEL. WSDL é uma tecnologia bem conhecida e com muitos usos além de definição de processos. Arquivos com essa extensão especificam a interface do *Web Service*, as propriedades, operações, bem como serviços implementados e chamados pelo processo. Arquivos com extensão BPEL representam a definição do processo, incluindo atividades, correlações, variáveis e eventos.

A Tabela 2.4 descreve os elementos de um arquivo de extensão BPEL.

Tabela 2.4 – Elementos do BPEL

| Nome  | Descrição   |
|---|---|
| <i>Process</i>                              | Representa um processo de negócio pode que conter um ou mais objetos.   |
| <i>Variable</i>                             | Uma variável para o uso em um processo ou um escopo, com o tipo baseado em um tipo de mensagem WSDL, um elemento XSD ou um tipo básico XSD. |
| <i>Property, Property Alias (from WSDL)</i> | Uma propriedade é um dado de uma mensagem WSDL. Um <i>property alias</i> é uma expressão XPath para encontrar o valor de uma propriedade.   |
| <i>Correlation Set</i>                      | Conjunto de uma ou mais propriedades usadas para relacionar processos que se comunicam.   |
| <i>Partner Link Type (from WSDL)</i>        | Mapeamento de tipos de portas de um <i>web service</i> com papéis.  |
| <i>Partner Link</i>                         | Definição das ligações entre processos, descrevendo qual papel cada um deles deve efetuar.  |
| <i>Partner</i>                              | Um conjunto de <i>Partner Links</i> .   |
| <i>Compensation Handler</i>                 | Usado quando um processo que já terminou tem que ser revertido para o estado inicial.   |
| <i>Fault Handler, Catch, CatchAll</i>       | Conjunto de <i>handlers</i> para tratar exceções em processos.  |
| <i>EventHandler, onMessage, onAlarm</i>     | Conjunto de <i>handlers</i> usados para processar eventos não solicitados.  |
| <i>Activity</i>                             | Um processo ou escopo tem exatamente uma atividade.   |
| <i>Receive</i>                              | Recebe uma mensagem SOAP.   |
| <i>Invoke</i>                               | Chama um <i>web service</i> parceiro, de forma síncrona ou assíncrona.  |
| <i>Reply</i>                                | Retorna uma resposta síncrona para um <i>web service</i> .  |
| <i>Compensate</i>                           | Uma atividade que ativa uma compensação.  |
| <i>Throw</i>                                | Uma atividade que pode gerar falha, ativando o <i>fault handler</i> daquele processo ou escopo.   |
| <i>Assign</i>                               | Copia os dados de uma variável para outra.  |
| <i>Wait</i>                                 | Pausa o processo por um tempo específico.   |
| <i>Empty</i>                                | Não faz nada.   |
| <i>Switch</i>                               | Uma estrutura de ou exclusivo, executa a atividade que a condição for verdadeira.   |
| <i>Flow</i>                                 | Estrutura de execução paralela.   |
| <i>Sequence</i>                             | Executa um conjunto de atividades sequencialmente.  |
| <i>Pick</i>                                 | Espera por um ou vários eventos acontecerem. Executa a atividade correspondente ao primeiro elemento que é ativado.                         |
| <i>While</i>                                | Executa uma atividade repetidamente enquanto a expressão for verdadeira.  |
| <i>Scope</i>                                | Uma atividade com o seu conjunto de <i>handlers</i> , variáveis e <i>correlation sets</i> .   |

Grandes companhias deram origem e apóiam o BPEL, sendo uma linguagem bastante aceita na indústria, e bem suportada por ferramentas. Como a especificação do BPEL não define uma notação gráfica, várias dessas ferramentas disponibilizam sua própria representação gráfica dos elementos. O objetivo disso é facilitar a construção do processo na forma de um diagrama que, posteriormente é convertido pela ferramenta para o formato definido na especificação do BPEL. Para fins de exemplificação, será mostrado um pouco do *BPEL Designer*, presente no *Oracle BPEL Process Manager*.

*BPEL Process Manager* é um ambiente de execução da *Oracle* para processos BPEL. É dividido em quatro partes principais: BPEL Server (onde os processos executam e se comunicam), BPEL Console (permite instalar, gerenciar e detectar erros dos processos), Database (banco de dados que armazena os esquemas BPEL) e BPEL Designer (permite desenvolver os processos BPEL de forma gráfica, sem a necessidade de escrever código).

BPEL Designer está disponível para dois ambientes de desenvolvimento, o *JDeveloper* e o *Eclipse*. No *Eclipse*, deve-se instalar-se um *plugin* para usá-lo. A Figura 2.4 mostra a modelagem de um processo no BPEL Designer para Eclipse. Nela, pode-se observar à esquerda do diagrama, uma lista com vários dos elementos descritos na Tabela 2.4. Esses elementos podem ser arrastados para a janela onde o diagrama é desenhado, permitindo a modelagem gráfica do processo.

O Anexo 1 apresenta um arquivo de extensão BPEL com o objetivo de exemplificar o uso das estruturas apresentadas na Tabela 2.4.

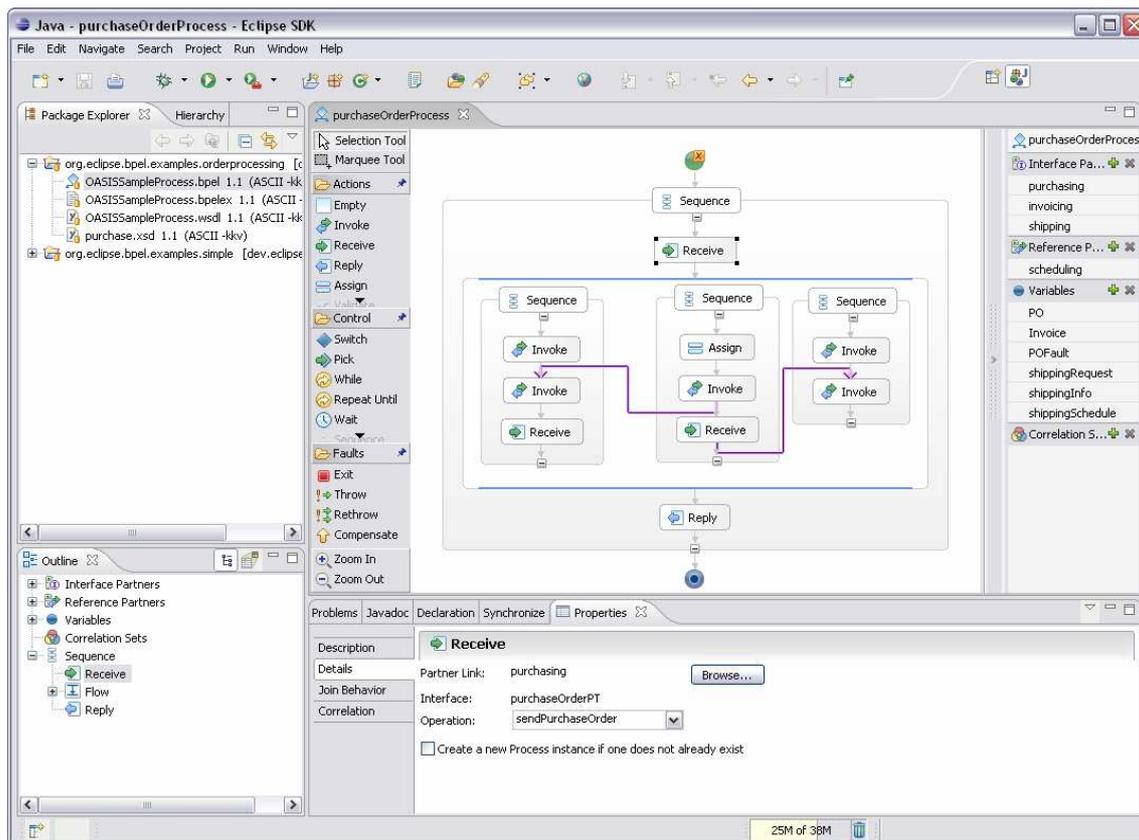


Figura 2.4 – Exemplo de criação de um processo BPEL na plataforma Eclipse

### 2.3.2 WfMC Reference Model

O modelo de referência da WfMC (*Workflow Management Coalition*) descreve em alto nível as partes principais de um sistema de gerenciamento de *workflow* (HAVEY, 2005 apud HOLLINGSWORTH, 2004).

A WfMC foi fundada em 1993 e é uma organização sem fins lucrativos que escreve padrões de *workflow*, incluindo o seu modelo de referência (HOLLINGSWORTH, 1995) e a especificação dos elementos que o constituem. A WfMC também produz bastante literatura sobre BPM. São membros desse grupo mais de 300 organizações, como por exemplo, IBM, Oracle e Sun (WfMC, 2009).

A estrutura do modelo de referência está representada na figura 2.5. As caixas representam componentes, e as flechas interfaces.

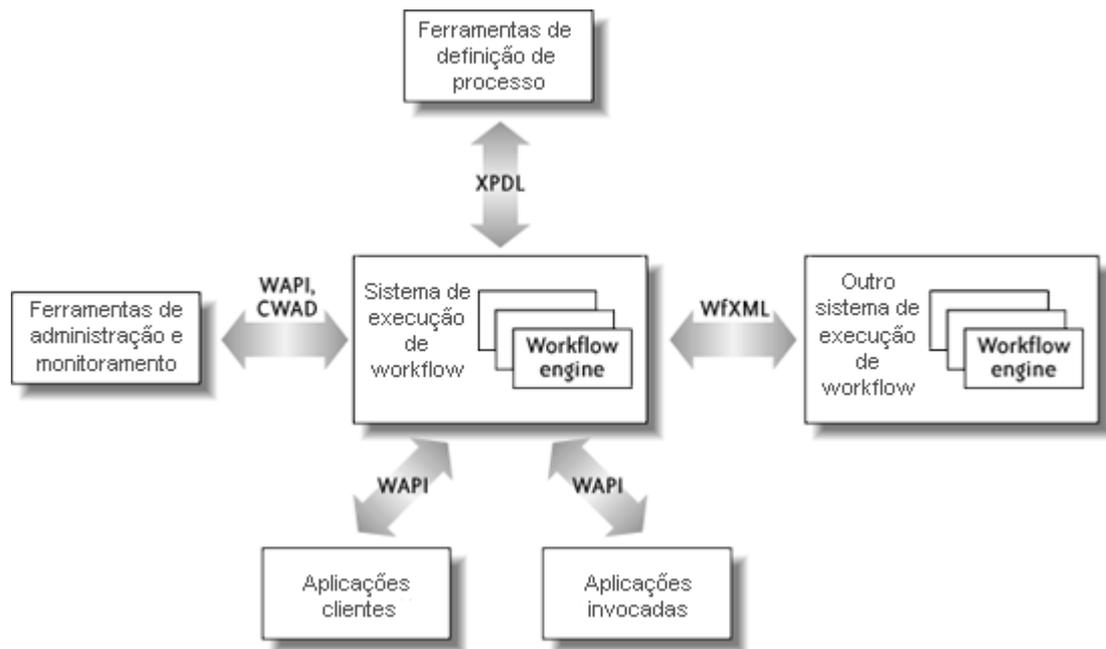


Figura 2.5 – Modelo de referência da WfMC (HAVEY, 2005)

Compõem o modelo de referência a especificação das interfaces: a) XPDL (*XML Process Definition Language*, utilizada para definir processos em formato baseado em XML); b) WAPI (*Workflow Application Programming Interface*, uma API para interação entre o sistema de execução de *workflow* e outras aplicações); e c) WfXML (*Workflow XML*), uma linguagem baseada em *web services* para a comunicação entre sistemas de execução de *workflow*.

Esse trabalho somente da XPDL, pois o foco dessa seção é execução de processos.

XPDL é uma interface entre a ferramenta de definição de processo e o serviço de execução de *workflow*. Os processos criados usando a ferramenta são exportados no formato XPDL e carregados em um serviço para a execução. Outros usos incluem a troca de definição de processos entre ferramentas, as quais costumam ter a função de exportar e importar arquivos nesse formato (HAVEY, 2005)

A WfMC não define um padrão gráfico de notação, e portanto, a aparência de um processo depende da ferramenta utilizada.

A Figura 2.6 trás um diagrama de classes que representa a estrutura do XPDL e a Tabela 2.5 esclarece os principais elementos presentes em um arquivo XPDL com uma breve descrição.

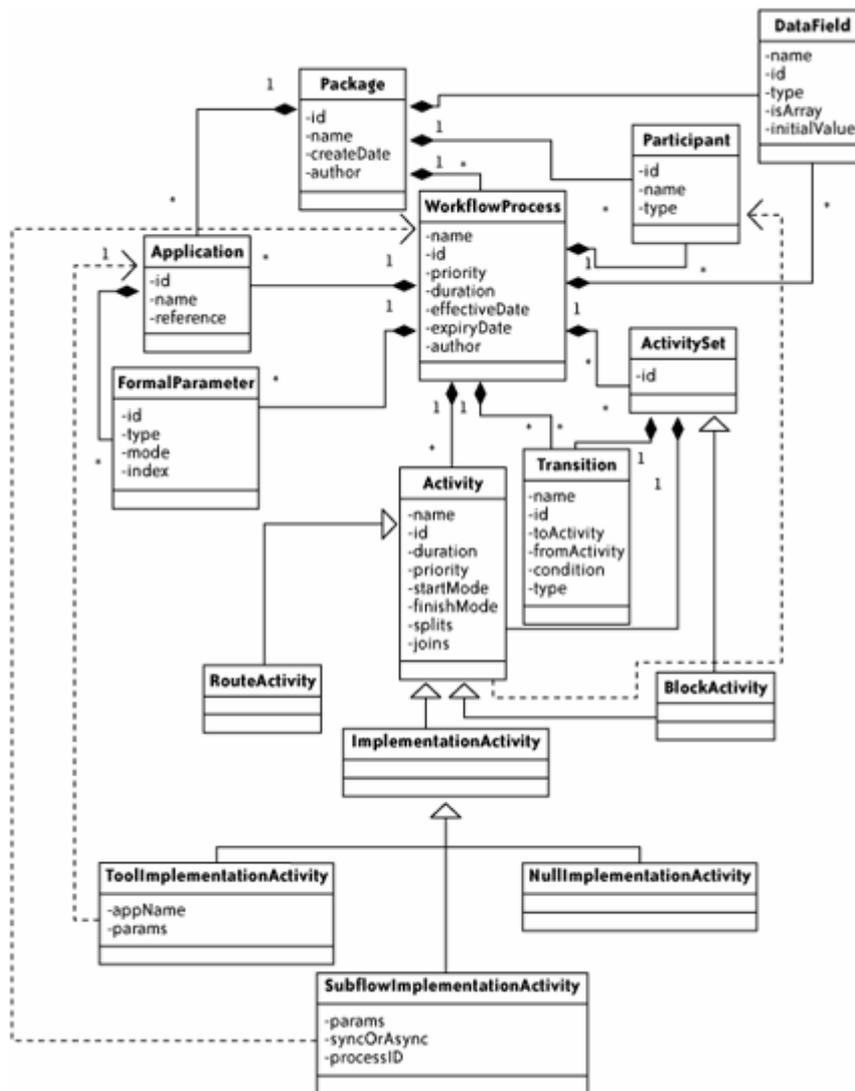


Figura 2.6 – Diagrama de classes do XPD (HAVEY, 2005)

Tabela 2.5 – Elementos do XPD

| Nome                   | Descrição   |
|------------------------|---|
| <i>Package</i>         | A entidade de mais alto nível. Define a aplicação inteira ou um grande sistema e consiste de uma ou mais definições de processos. |
| <i>DataField</i>       | Dados relevantes ao workflow  |
| <i>Participant</i>     | Um ator que realiza um papel chave no processo. Pode ser humano ou um sistema.  |
| <i>Application</i>     | Componente programático chamado por um processo.  |
| <i>WorkflowProcess</i> | Estrutura central do XPD consiste de uma ou mais atividades e um conjunto de transições.  |
| <i>Transition</i>      | Um caminho entre uma atividade e outra.   |
| <i>FormalParameter</i> | Um dado de entrada.   |
| <i>RouteActivity</i>   | Direciona o fluxo para as transições de saída.  |
| <i>BlockActivity</i>   | Conjunto de atividades que tem o seu próprio escopo,  |

|                                      |   |
|--------------------------------------|---|
|                                      | semelhante ao scope do BPEL.  |
| <i>ToolImplementationActivity</i>    | Uma chamada de uma aplicação  |
| <i>NullImplementationActivity</i>    | Uma atividade manual, designada para um participante humano.            |
| <i>SubflowImplementationActivity</i> | Chama outro processo de <i>workflow</i> , como um subprocesso aninhado. |

O exemplo a seguir foi retirado de HAVEY (2005) e representa um processo de negócio de uma agência de viagens modelado em XPDL. O agente de viagens recebe o itinerário desejado pelo cliente e realiza de forma paralela, as reservas de hotéis, aluguel de carros e programação de vôos. Se tudo ocorrer bem, o agente envia uma confirmação por e-mail ao cliente; caso contrário (se alguma dessas tarefas falhar), o agente solicita ao cliente mudanças no itinerário. O processo foi modelado usando a ferramenta *open source* JAWE (Enhydra Java Workflow Editor). A Figura 2.7 mostra o programa em uso.

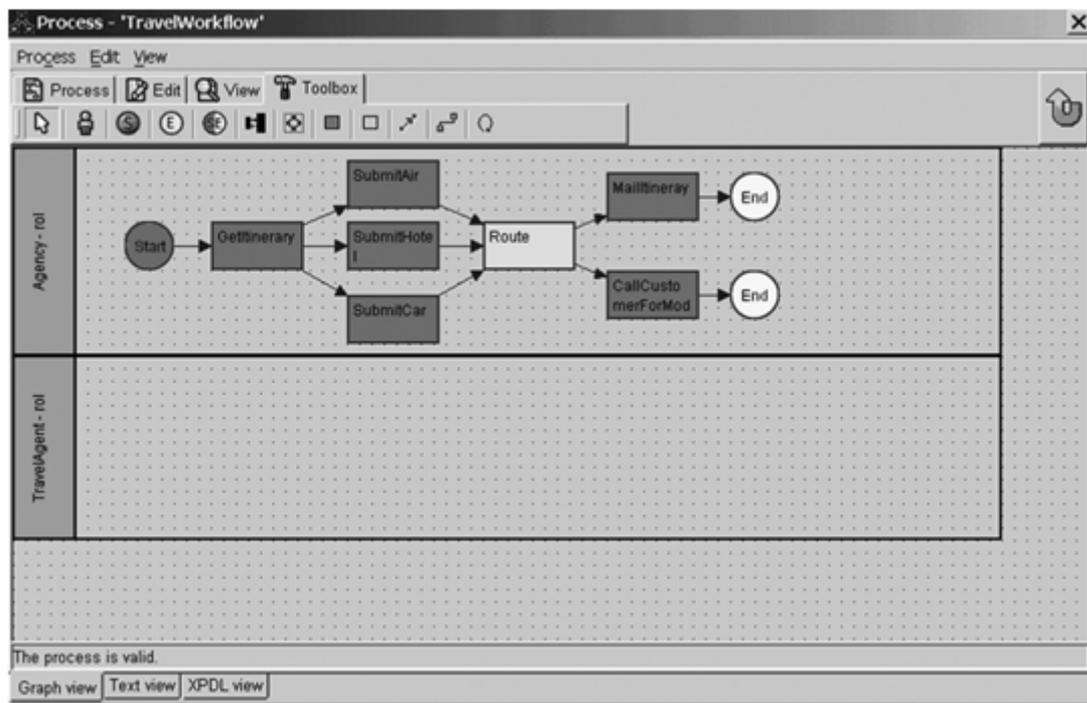


Figura 2.7 – Exemplo de uso do programa JAWE (HAVEY, 2005)

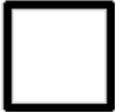
O código gerado por essa ferramenta, juntamente com uma breve explicação, pode ser encontrado no Anexo 2.

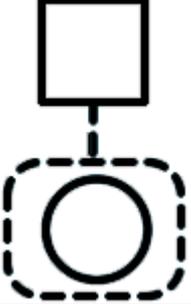
### 2.3.3 YAWL

YAWL (Yet Another Workflow Language) é uma linguagem de definição de *workflow* proposta por AALST e HOFSTEDE (2003) e baseada em um estudo em que esses autores levantaram 20 padrões de *workflow* recorrentemente encontrados (AALST et al, 2003). Com base nesses padrões, foi feita uma comparação da funcionalidade de 15 sistemas de gerenciamento de *workflow*, concluindo-se que nenhum deles apoiava, de forma satisfatória (com um esforço de modelagem aceitável), todos os padrões levantados. Essa deficiência motivou a criação do YAWL, que tomou inspiração nas redes de Petri, por várias razões, dentre elas, a semântica formal, o fato de ser baseada em estados e a abundância de técnicas de análise existentes.

Uma especificação de *workflow* em YAWL é um conjunto de redes EWF (*Extended Workflow Nets*) (AALST e HOFSTEDE, 2003) que formam uma hierarquia. Cada rede EWF é formada por tarefas atômicas ou compostas e condições que podem ser interpretadas como lugares, e tem uma única condição de entrada e uma única condição de saída. A Tabela 2.6 descreve os elementos usados no YAWL.

Tabela 2.6 – Elementos do YAWL

| Elemento                  | Descrição   | Notação   |
|---------------------------|---|---|
| Tarefa atômica            | Uma única tarefa realizada por uma pessoa ou uma aplicação externa. |  |
| Condição de entrada       | Onde o processo começa.   |  |
| Divisão do tipo E lógico  | Ativa todas as saídas dessa tarefa ao ser concluída.                |  |
| Divisão do tipo Ou lógico | Ativa um número de saídas dessa tarefa ao ser concluída.            |  |

|                                |  |   |
|--------------------------------|--|---|
| Divisão do tipo Ou exclusivo   | Ativa uma saída dessa tarefa ao ser concluída.   |    |
| Tarefa composta                | Um container para outros processos YAWL, permitindo decomposição de diagramas.   |    |
| Condição                       | Representa um estado de um processo.   |    |
| Condição de saída              | Onde o processo termina.   |    |
| Junção do tipo E lógico        | Ativa essa tarefa quando todas as entradas são ativadas.   |    |
| Junção do tipo OU lógico       | Ativa essa tarefa quando um ou mais entradas são ativadas e não há possibilidade para outras entradas serem ativadas se a tarefa continua a esperar.                         |   |
| Junção do tipo OU exclusivo    | Ativa essa tarefa cada vez que uma entrada é ativada.  |  |
| Tarefa de múltiplas instâncias | Permite múltiplas instâncias de uma mesma tarefa executando de forma concorrente. O número mínimo e máximo de instâncias e outros parâmetros podem ser especificados.        |  |
| Região de cancelamento         | Todos os elementos dentro da região pontilhada são desativados quando a tarefa é ativada. É possível especificar o cancelamento de tarefas simples ou de processos inteiros. |  |

O diagrama presente na Figura 2.8 representa o pagamento em uma empresa, e utiliza várias das estruturas apresentadas na Tabela 2.6.

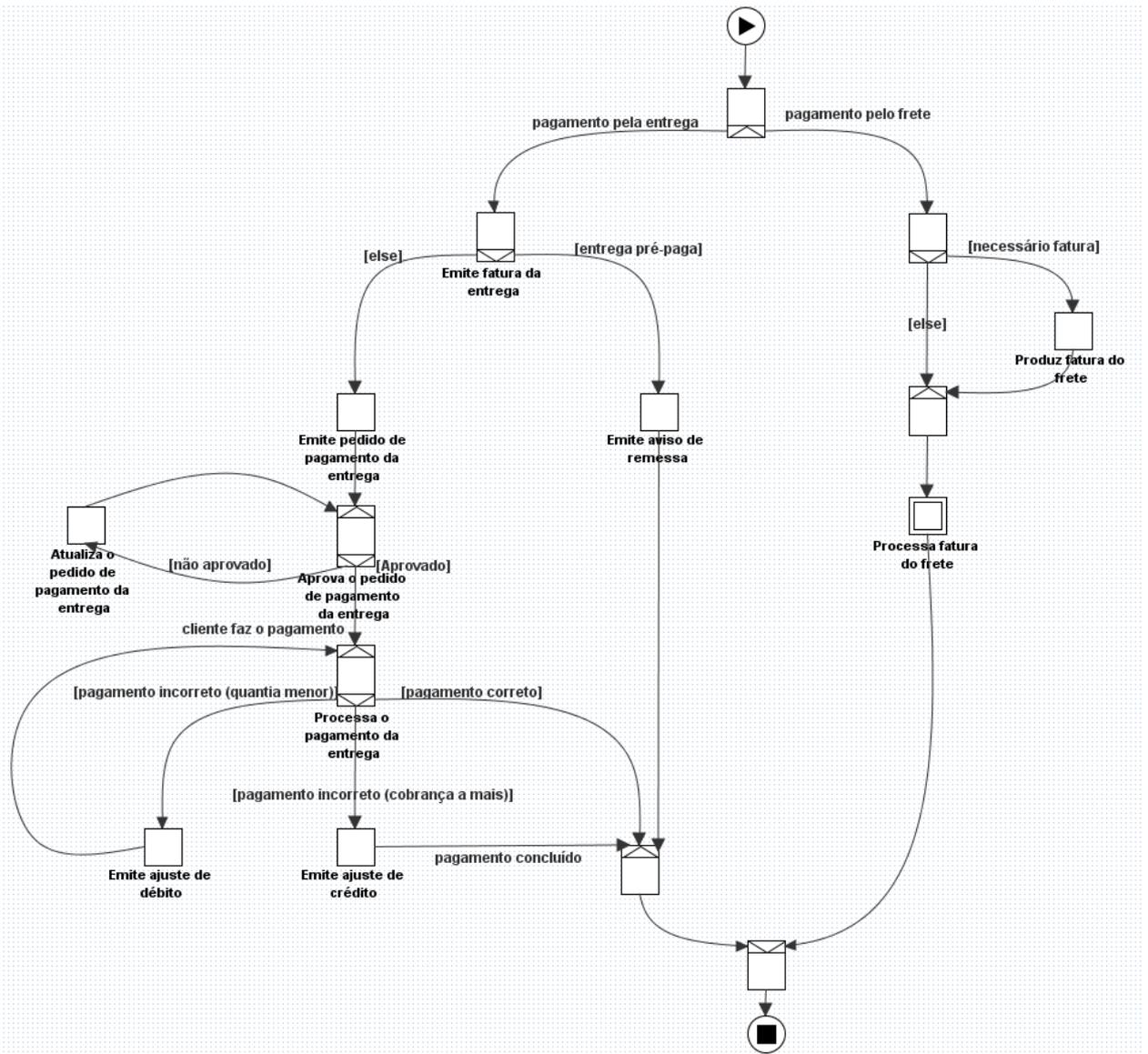


Figura 2.8 – Exemplo de uso de YAWL (AALST e HOFSTEDDE, 2003)

O processo começa com uma divisão do tipo e lógico, em que ambos os caminhos são seguidos, ou seja, ocorre o "pagamento pela entrega" e o "pagamento pelo frete". O "pagamento pelo frete" segue para uma divisão do tipo ou exclusivo, onde apenas um caminho é seguido. Caso seja "necessário fatura", ela é produzida; caso contrário, nada acontece. Os dois caminhos seguem para uma junção do tipo *ou exclusivo*, que segue para uma tarefa composta (não representada nesse diagrama) que "processa a fatura do frete". O fluxo segue para a junção do tipo *e lógico* que leva ao final do processo.

O "pagamento pela entrega" começa com uma divisão do tipo *ou exclusivo*; se for do tipo "entrega pré-paga", o fluxo segue para uma tarefa atômica que "emite aviso

de remessa" e segue para uma junção do tipo *ou exclusivo*. Senão, acontece uma tarefa atômica que "emite pedido de pagamento da entrega". Prossegue-se então para uma junção do tipo *ou exclusivo* representada em conjunto com uma divisão do tipo *ou exclusivo*. Essa estrutura tem a função de avaliar o pedido de pagamento: se ele não for aprovado, ocorre uma tarefa que propõe mudanças no pedido, que volta para a avaliação. Se ele for aprovado, o fluxo segue para outra estrutura semelhante, que processa o pagamento do pedido. Se o pagamento realizado for menor que o valor da fatura, ocorre a tarefa que "Emite ajuste de débito"; se foi cobrado mais do que o valor da fatura, ocorre a tarefa "Emite ajuste de crédito" e segue para a junção do tipo *ou exclusivo*. Caso o pagamento esteja correto, o fluxo também segue para a junção do tipo *ou exclusivo*. Após essa junção, existe uma estrutura de junção do tipo *e lógico*. Essa estrutura significa que tanto o caminho seguido por "pagamento pela entrega" quanto o caminho seguido por "pagamento pelo frete" foram concluídos. Só então o processo acaba.

## **2.4 Modelagem de organização de negócio**

Uma organização é um sistema de recursos que procura realizar um objetivo ou conjuntos de objetivos. Um sistema é um todo complexo e organizado, formado de partes ou elementos que interagem para realizar um objetivo explícito. Todas as organizações são sistemas, embora nem todos os sistemas sejam organizações. As pessoas são o principal recurso das organizações (MAXIMIANO, 2000).

Segundo BRIDGELAND e ZAHAVI (2009), modelar organizações é descrever através de modelos os departamentos, os grupos de trabalho, as interações entre pessoas, os papéis representados por essas pessoas e as interações da organização com outras organizações.

Modelar a organização é útil, pois facilita a comunicação, o treinamento e a análise. A análise de um modelo de organização permite, por exemplo, remanejar recursos humanos com maior facilidade. Esse tipo de modelo também facilita a comunicação dessas mudanças aos funcionários. Em relação ao treinamento, um funcionário novo, ao visualizar um diagrama da organização, compreende rapidamente a hierarquia e os papéis desempenhados na organização. Essa seção tratará dos diagramas utilizados para modelar organizações.

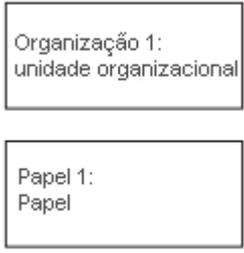
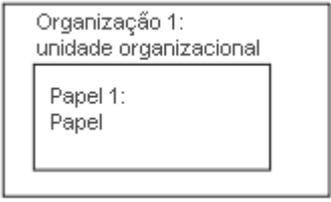
## 2.4.1 Relationship Maps

Segundo BRIDGELAND e ZAHAVI (2009), não existe um padrão para modelagem de organização. Cada ferramenta de modelagem tem uma abordagem própria e dessa maneira existe muita variedade. Ainda segundo os autores, o que mais se aproxima de um padrão, com muitos praticantes e muitas variantes, é o *Relationship Map*.

*Relationship Maps* representam as organizações e os papéis como retângulos, que podem estar contidos em outros retângulos indicando que um componente é parte de outro. Setas conectam os retângulos, representando as interações entre as organizações ou papéis. O *Relationship Map* também permite que se inclua um organograma dentro de um retângulo, representando a hierarquia da organização.

A Tabela 2.7 descreve os elementos presentes em um *Relationship Map* e o exemplo da Figura 2.9 ilustra o seu uso.

Tabela 2.7 – Elementos do *Relationship Map*

| Elemento                  | Descrição  | Notação   |
|---------------------------|--|---|
| Organização ou papel      | Um retângulo pode representar tanto uma organização (conjunto de pessoas trabalhando em conjunto em direção a um objetivo comum) ou um papel (responsabilidade que uma pessoa assume enquanto ela possui um cargo na organização). |  |
| Associação: parte de      | Um retângulo dentro de outro indica que um elemento pertence a outro. Um papel dentro de uma organização, por exemplo, indica que existe uma pessoa dentro da organização com aquele papel.  |  |
| Associação: subordinado a | Um organograma representa a hierarquia de uma organização, e pode ser incluído dentro de um retângulo.   |  |
| Associação: Interação     | Mostra quem trabalha com quem. Esse tipo de associação possui um pequeno texto que indica a contribuição associada com a interação, por exemplo, um produto ou serviço.  |  |
| Associação: Influência    | Representa um relacionamento indireto, como no caso de duas organizações (ou   |  |

|  |  |  |
|--|--|--|
|  | papéis) que não trabalham juntas, mas as ações de uma podem causar efeitos em outra. |  |
|--|--|--|

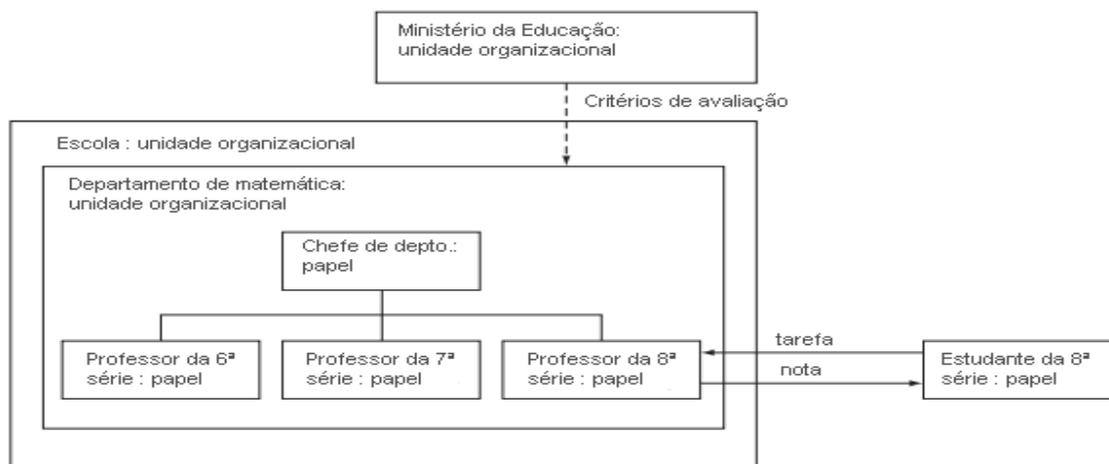


Figura 2.9 – Exemplo de uso de *Relationship Map* (BRIDGELAND e ZAHAVI, 2009),

## 2.5 Modelagem de regras de negócio

De acordo com BRG (2000), regra de negócio é uma declaração que define ou restringe algum aspecto do negócio, com a intenção de afirmar a estrutura, controlar ou influenciar seu comportamento.

Regras são um aspecto importante do negócio, na medida em que capturam restrições de comportamento. Isso significa que elas definem o que é permitido e o que não é dentro de uma organização. Esse conhecimento é útil principalmente para especificar os requisitos de software de uma aplicação que vai ser desenvolvida. Segundo BRIDGELAND e ZAHAVI (2009), o principal motivo para criar um modelo de regras de negócio atualmente é apoiar a etapa de especificação de requisitos.

Além do motivo anterior, modelos de regras de negócio são úteis na comunicação e no treinamento, já que oferecem precisão na definição da regra, evitando ambigüidades. Um modelo de regras também apóia a gerência de conhecimento, capturando e preservando o comportamento da organização.

A seguir são apresentadas duas formas de modelagem de regras de negócio: RuleSpeak e SBVR. O RuleSpeak foi muito utilizado antes do surgimento do SBVR, que se tornou um padrão especificado e mantido pela OMG

### 2.5.1 RuleSpeak

*RuleSpeak* foi desenvolvido em 1996 por Ron Ross e seus colaboradores da *Business Rules Solutions* (RULESPEAK, 2009). É um estilo de se escrever regras, não é uma linguagem. Fornece uma série de orientações para se expressar regras de negócio de forma concisa e não-ambígua. A versão original é em inglês, mas existem traduções para espanhol, holandês e alemão. *RuleSpeak* foi bastante utilizado durante dez anos e foi uma das três notações utilizadas como referência na criação do padrão SBVR.

Uma frase que descreve uma regra é composta por termos, fatos e palavras-chaves. O *RuleSpeak* propõe que as regras sejam expressas em formas sentenciais e usem palavras-chave definidas, garantindo concisão e evitando ambigüidade. Algumas formas sentenciais e palavras-chave usadas no *RuleSpeak* são listadas na Tabela 2.8

Tabela 2.8 – *RuleSpeak*

| Forma Sentencial                             | Palavras-chave                         |
|--|--|
| Forma sentencial de obrigação                | <i>r must s</i>                        |
| Forma sentencial de proibição                | <i>r must not s</i>                    |
| Forma sentencial de permissão restrita       | <i>r may s only t</i>                  |
| Forma sentencial de permissão irrestrita     | <i>r may s</i><br><i>r need not s</i>  |
| Forma sentencial de necessidade              | <i>r always s</i>                      |
| Forma sentencial impossibilidade             | <i>r never s</i>                       |
| Forma sentencial de possibilidade restrita   | <i>r can s only t</i>                  |
| Forma sentencial de possibilidade irrestrita | <i>r sometimes s</i><br><i>r can s</i> |

Um exemplo de regra escrita com *RuleSpeak* é dado a seguir:

**Each** rental car **must** *be owned by* **exactly one** branch.

A regra diz que cada carro disponível para aluguel deve pertencer a apenas uma filial. As palavras em negrito são palavras-chave: “must” indica obrigatoriedade, “each” e “exactly one” são quantificadores. As palavras sublinhadas são termos, geralmente nomes relativos à atividade que o negócio desempenha. As palavras em itálico indicam

um fato. Quantificadores, nome e fatos serão descritos com mais detalhes na Seção 2.5.2.

## 2.5.2 SBVR

O *Semantics of Business Vocabulary and Business Rule* (SBVR) (OMG, 2008b), é um padrão que permite definir regras de negócio e o vocabulário associado a elas. Trata-se de um inglês estruturado, com um conjunto de palavras-chave usadas na construção de frases que expressam regras de negócio. A especificação do SBVR é extensa, e nesse trabalho serão tratados apenas os elementos mais úteis para o cotidiano da modelagem de negócio, seguindo a abordagem de BRIDGELAND e ZAHAVI (2009).

Uma sentença que expressa uma regra possui nomes, fatos e palavras-chave. Nomes são substantivos que representam objetos presentes no negócio. Fatos caracterizam como os nomes se relacionam. O significado de cada nome e fato deve ser definido. Palavras-chave tem o significado definido na própria especificação do SBVR.

O vocabulário é expresso com uma série de conceitos, como por exemplo:

pagamento

Definição: Ação ou efeito de pagar, de saldar.

Fonte: Dicionário Aurélio Online

dinheiro

Definição: A moeda corrente.

Fonte: Dicionário Aurélio Online

pagamento somente em dinheiro

Definição: Pagamento que envolve dinheiro.

Conceitos são definidos com o uso de dicionários, ou construídos tendo por base conceitos previamente definidos.

As palavras-chave são divididas em quantificação, operações lógicas e operações modais. A Tabela 2.9 descreve o significado das palavras-chave do SBVR. As letras “m” e “n” representam números, e “p” e “q” proposições lógicas.

Tabela 2.9 – Palavras chaves do SBVR (OMG, 2008b)

| Quantificadores          |   | Operações Lógicas         |                     |
|--------------------------|---|---------------------------|---------------------|
| Palavra-chave            | Significado   | Palavra-chave             | Significado         |
| each                     | Quantificação universal                                     | it is not the case that p | Negação lógica      |
| some                     | Quantificação existencial                                   | p and q                   | Conjunção           |
| at least one             | Quantificação de no mínimo 1                                | p or q                    | Disjunção           |
| at least n               | Quantificação de no mínimo n                                | p or q but not both       | Disjunção exclusiva |
| at most one              | Quantificação de no máximo 1                                | if p then q               | Implicação          |
| at most n                | Quantificação de no máximo n                                | q if p                    | Implicação          |
| exactly one              | Quantificação de exatamente 1                               | p if and only if q        | Equivalência        |
| exactly n                | Quantificação de exatamente n                               | not both p and q          | Formulação nand     |
| at least n and at most m | Quantificação de intervalo numérico                         | neither p nor q           | Formulação nor      |
| more than one            | Quantificação de 1 ou mais                                  | p whether or not q        | Formulação ou não   |
| Operações modais         |   | Outras                    |                     |
| Palavra-chave            | Significado   | Palavra-chave             |                     |
| it is obligatory that p  | Formulação de obrigação                                     | a, an                     |                     |
| it is prohibited that p  | Formulação de obrigação em conjunto de uma negação lógica   | another                   |                     |
| it is necessary that p   | Formulação de necessidade                                   | a given                   |                     |
| it is impossible that p  | Formulação de necessidade em conjunto de uma negação lógica | that                      |                     |
| it is possible that p    | Formulação de possibilidade                                 | who                       |                     |
| it is permitted that p   | Formulação de permissibilidade                              | is of                     |                     |
|                          |   | what                      |                     |

A seguir, são apresentadas algumas regras definidas utilizando o inglês estruturado proposto pelo SBVR, para exemplificação. As palavras em negrito são palavras-chave, as sublinhadas são nomes (que devem ser definidos no vocabulário), e as palavras em itálico são fatos. Essas regras se aplicam a uma companhia de aluguel de carros e fazem parte de um exemplo disponível em (OMG, 2008b).

**It is obligatory that each rental car is owned by **exactly one** branch.**

**It is obligatory that the rental duration of each rental is at most 90 rental days.**

**A rental must have at most three additional drivers.**

**The rental report of each rental must specify **what car** is assigned to **the rental**.**

A primeira regra diz que é obrigatório que cada carro disponível para locação pertença a exatamente uma filial. A segunda, que cada locação pode durar no máximo 90 dias. A terceira especifica que cada locação tenha, no máximo, três motoristas adicionais. A quarta diz que o relatório de cada locação deve obrigatoriamente especificar qual carro foi locado. Todos os nomes (palavras sublinhadas) devem estar definidos no vocabulário (omitido aqui).

## **2.6 Modelagem de motivação de negócio**

Segundo a OMG (2008c), motivação de negócio trata das razões por trás de cada abordagem escolhida para uma atividade do negócio e dos resultados que ela pretende alcançar.

Basicamente, modelos de motivação de negócio capturam o que a organização está tentando fazer (objetivos) e como ela faz (estratégias). Também procuram representar fatores que podem influenciar nessas escolhas (influenciadores).

Modelos de motivação de negócio são úteis, pois podem ser usados para comunicar os objetivos e estratégias de uma organização para seus empregados atuais, bem como serem usados em treinamentos de novos empregados. Eles também auxiliam na gerência do conhecimento, pois documentam as aspirações da organização. Além disso, podem ser analisados para a comparação de estratégias.

Antes da adoção do BMM como um padrão pela OMG, havia muita variedade de técnicas usadas para modelar os objetivos e estratégias de uma organização (BRIDGELAND e ZAHAVI, 2009). Esse trabalho abordará apenas o BMM, na seção seguinte.

### **2.6.1 BMM**

O *Business Motivation Model* (OMG, 2008c) é um padrão para modelagem de motivação de negócio que provê um esquema ou estrutura para o desenvolvimento, comunicação e gerência de planos de negócio de uma maneira organizada. Basicamente

ele permite identificar os fatores que motivam o estabelecimento de planos de negócio, os elementos presentes nesses planos e sua inter-relação. BMM trabalha com a idéia da motivação presente em uma organização, suas aspirações e as ações que serão tomadas para satisfazê-las. BMM também se preocupa em representar os influenciadores, quem podem causar impacto no trabalho da organização.

O diagrama apresentado na Figura 2.10 provê uma visão geral do BMM, representando os três aspectos mencionados de uma organização: aspirações (fins), ações tomadas para satisfazê-las (meios), e os influenciadores.

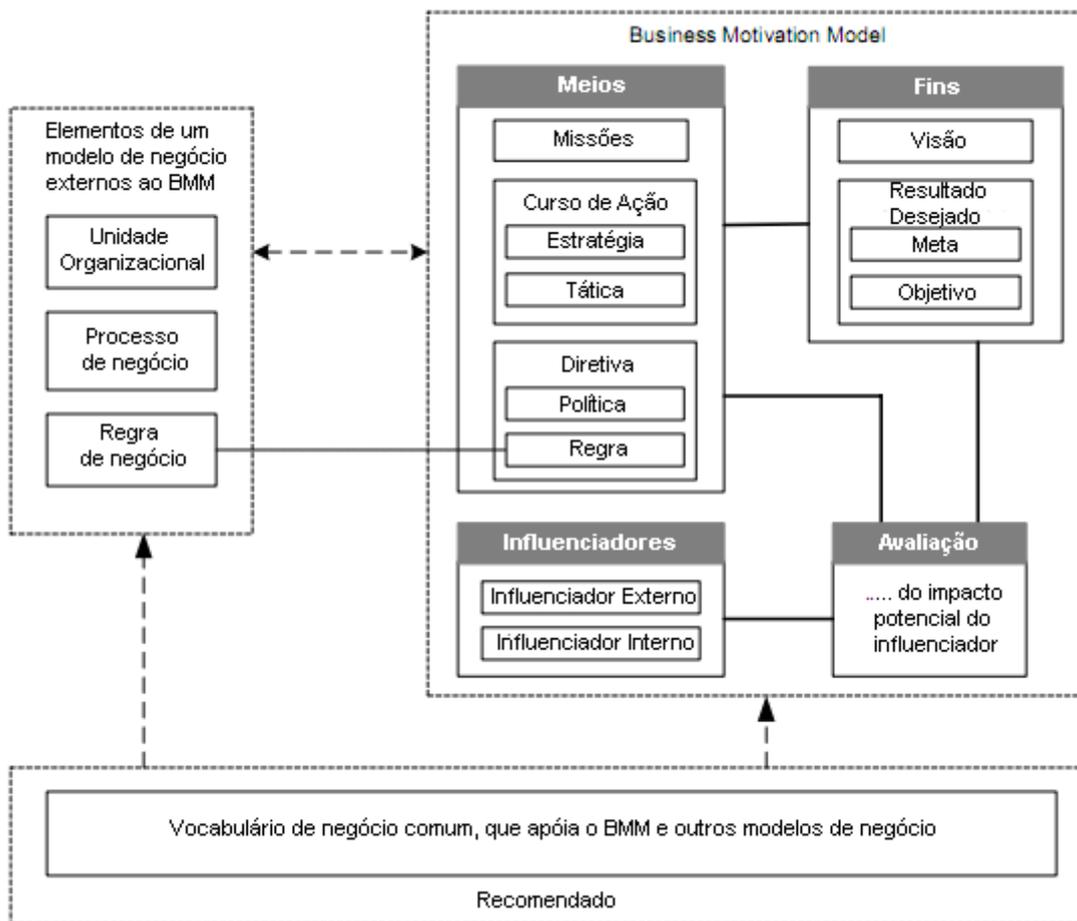


Figura 2.10 – Estrutura do BMM (OMG, 2008c)

Os fins tratam do que uma organização quer ser. Podem tratar de mudanças que ela deseja, ou simplesmente a manutenção do que ela já tem no mercado competitivo. Um fim não descreve como isso é conseguido. Os fins se dividem em visão e em resultados desejados. Uma visão é um panorama geral do que a empresa deseja, ao passo que o resultado desejado é mais específico, definindo cada meta e objetivo que

deve ser cumprido. Metas são aspirações focalizadas, e objetivos são metas quantificadas e caracterizadas com tempo.

Os meios tratam do que uma organização decidiu fazer para se tornar o que deseja ser. São divididos em missão, curso de ação e diretiva. Missão indica a atividade operacional que já ocorre na organização. É possível definir um modelo de motivação de negócio sem definir explicitamente a missão. Curso de ação define o que tem que ser feito (mas não define como será feito) e pode ser uma tática (geralmente pequena e de escopo limitado) ou uma estratégia (geralmente maior e com um escopo grande). Diretivas podem ser regras de negócio ou políticas. Regras de negócio já foram definidas na Seção 2.5, e as políticas têm um papel parecido, porém geralmente são menos formais e fornecem apenas orientações.

Influenciador é algo que causa mudanças ou afeta a organização no emprego dos seus meios e no alcance de seus fins. Podem ser internos ou externos a influência pode ser negativa ou positiva. A avaliação da influência é uma etapa importante para que sejam feitas as mudanças necessárias nos fins e nos meios.

A Tabela 2.10 apresenta alguns exemplos de cada elemento do BMM descrito para o contexto uma pizzaria.

Tabela 2.10 – Exemplo de uso do BMM

| <b>Elemento</b> | <b>Exemplo</b>  |
|-----------------|---|
| Visão           | Ser a pizzaria mais popular da cidade   |
| Meta            | Entregar pizzas dentro do período de tempo estipulado.  |
| Objetivo        | Até Janeiro de 2010, 95% das pizzas entregues no período de tempo estipulado.                 |
| Missão          | Entregar pizzas na cidade toda.   |
| Estratégia      | Entregar pizzas na localidade de escolha do cliente.  |
| Tática          | Contratar motoristas que façam entregas com seus próprios veículos.                           |
| Política        | Segurança na cozinha e nas ruas vem sempre em primeiro lugar.                                 |
| Regra           | Pizzas devem ser entregues no prazo máximo de 1 hora.   |
| Influenciador   | Chuvas além do normal causaram diminuição na colheita de trigo e aumento no preço da farinha. |
| Avaliação       | Aumento do preço da pizza para compensar o aumento do preço da farinha.                       |

## 2.7 Considerações Finais

Esse capítulo fez um apanhado das principais linguagens existentes para a modelagem de negócio. A escolha de quais linguagens abordar se deu baseada na literatura, principalmente em BRIDGELAND e ZAHAVI (2009) e HAVEY (2005). O estudo de cada linguagem, sempre que possível, tomou por base as especificações oficiais, mas frequentemente foi enriquecido por explicações e exemplos presentes em vários livros. A apresentação do conteúdo do capítulo seguiu uma divisão em processos (modelagem e execução), organização, regras e motivação.

Com relação à modelagem de processo de negócio, a linguagem que mais se destaca é o BPMN, por ter sido elaborada especialmente para isso. Quando se trata de execução de processos, BPEL é a mais usada, devido ao forte apoio de grandes organizações a essa linguagem. É possível mapear um processo escrito em BPMN para BPEL, como descrito em WHITE (2005).

Para modelagem de organização de negócio faltam padrões. A proposta de notação encontrada na literatura - *Relationship Maps*, foi descrita na Seção 2.41. Apesar de não ser um padrão, essa notação permite representar a organização com mais detalhes que um organograma, tradicionalmente utilizado para a mesma finalidade.

Para modelagem de regras de negócio são apresentadas duas abordagens. A primeira, o *RuleSpeak*, foi a mais utilizada durante dez anos, antes da criação do padrão SBVR. Devido a essa importância, e por ter influenciado diretamente a construção do SBVR, ele foi abordado brevemente na Seção 2.51. Atualmente, porém, SBVR é o padrão a ser usado quando se trata de regras de negócio.

Para modelagem da motivação, existe o padrão BMM, uma série de conceitos que permitem explicitar os objetivos e estratégias, bem como os influenciadores do negócio. É o padrão usado atualmente, embora muitas vezes enriquecido com uma representação gráfica.

No próximo capítulo é apresentado a modelagem do negócio de uma empresa real, utilizando-se as quatro linguagens destacadas nessa conclusão. Embora o capítulo anterior também tenha tratado de linguagens que permitem a execução de processos de negócio, tais linguagens não foram consideradas no exemplo descrito no próximo capítulo, por falta de tempo.

### **3. Modelagem de uma Empresa Real**

#### **3.1 Introdução**

O estudo realizado no Capítulo 2 teve objetivo de proporcionar uma visão panorâmica da modelagem de negócio, descrevendo suas principais notações e linguagens. No presente capítulo pretende-se fazer um exemplo de aplicação de algumas dessas linguagens para a criação de modelos de uma empresa real, em atividade atualmente. A modelagem a ser feita é da empresa júnior de informática da UFJF, a Base Três.

A Base Três Consultoria em Informática existe desde 1996 e é uma empresa sem fins lucrativos, constituída e administrada por alunos matriculados no curso de Ciência da Computação da Universidade Federal de Juiz de Fora. A empresa visa prestar serviços a preços acessíveis, de modo que pequenas empresas tenham acesso à consultoria de informática. Como utiliza mão de obra do próprio curso de Ciência da Computação, a empresa proporciona aos acadêmicos a inserção no mercado de trabalho e a utilização prática dos conhecimentos teóricos adquiridos em aula.

Para a realização da modelagem, foram feitas diversas reuniões com o diretor-presidente para a obtenção de uma descrição da empresa e disponibilização de documentos. A descrição e os documentos foram utilizados como base na construção dos modelos que, após concluídos, foram validados pelos membros da Base Três. As correções e modificações sugeridas foram feitas.

Os modelos foram construídos na ordem em que estão dispostos nesse capítulo. Essa ordem foi escolhida em função de complexidade crescente: o modelo de organização foi rapidamente construído, já os modelos de processos demandaram muito mais tempo. As sessões seguintes descrevem cada aspecto modelado.

#### **3.2 Modelagem da Organização**

Os modelos de organização foram descritos na Sessão 2.4. Eles objetivam descrever os departamentos, grupos de trabalho, interações entre pessoas, papéis representados por essas pessoas e interações da organização com outras organizações (BRIDGELAND e ZAHAVI, 2009).

Para a modelagem da estrutura da Base Três utilizou-se a descrição da equipe disponível no site da empresa (BASE TRÊS, 2009). Essa descrição apresenta os membros e os seus papéis, e a qual unidade organizacional eles pertencem. Após reunião com o diretor-presidente para o esclarecimento de dúvidas, foi possível construir o seguinte diagrama, utilizando um *Relationship Map*, conforme descrito na Sessão 2.41:

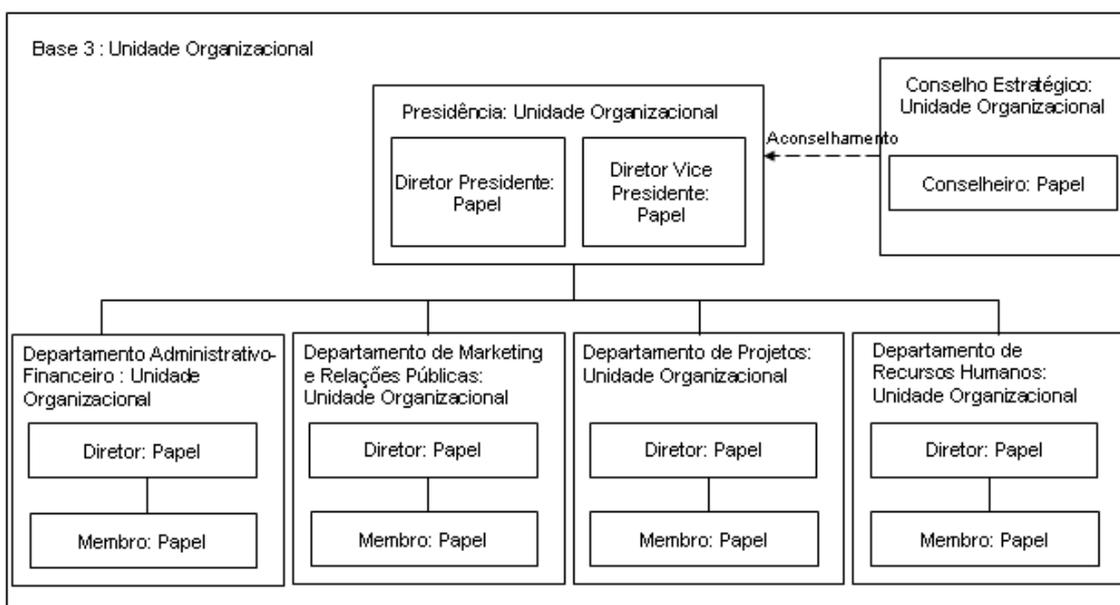


Figura 3.1 – *Relationship Map* da Base Três

A Base Três possui uma presidência e quatro departamentos. Cada departamento tem um diretor e membros subordinados a ele. O número de membros de cada departamento varia, podendo ser apenas um ou vários. Existe um Conselho Estratégico formado por pessoas que já participaram da Base Três, geralmente ex-presidentes ou ex-diretores, que funciona como órgão de aconselhamento da presidência.

### 3.3 Modelagem das Regras de Negócio

A modelagem de regras de negócio foi discutida na Sessão 2.5. Regras de negócio são frases que definem ou restringem algum aspecto do negócio, com a intenção de afirmar a estrutura, controlar ou influenciar o comportamento do negócio (BRG, 2000).

As regras de negócio da Base Três foram reescritas no padrão SBVR (OMG, 2008b) descrito na Sessão 2.5.2. Ainda, enquanto os documentos e membros da empresa utilizam a língua portuguesa, o padrão SBVR está baseado no inglês. Para não

abrir mão deste padrão e, ao mesmo tempo, facilitar a leitura das regras nele escritas, a Tabela 3.1 fornece uma tradução para o português, de cada palavra-chave do SBVR utilizada nas regras de negócio da Base Três.

Tabela 3.1 – Tradução das palavras chaves do SBVR

| Palavra-Chave Original  | Palavra-chave traduzida      |
|-------------------------|------------------------------|
| <b>It is permitted</b>  | <b>É permitido</b>           |
| <b>It is prohibited</b> | <b>É proibido</b>            |
| <b>It is obligatory</b> | <b>É obrigatório</b>         |
| <b>It is necessary</b>  | <b>É necessário</b>          |
| <b>It is possible</b>   | <b>É possível</b>            |
| <b>Only if</b>          | <b>Apenas se</b>             |
| <b>At most one</b>      | <b>No máximo 1</b>           |
| <b>All</b>              | <b>Todos</b>                 |
| <b>must</b>             | <b>Deve obrigatoriamente</b> |
| <b>Or</b>               | <b>Ou</b>                    |
| <b>And</b>              | <b>E</b>                     |
| <b>The</b>              | <b>O, a, os, as</b>          |
| <b>At least one</b>     | <b>Pelo menos 1</b>          |
| <b>Each</b>             | <b>Cada</b>                  |
| <b>That</b>             | <b>Que</b>                   |
| <b>If</b>               | <b>Se</b>                    |

As regras foram divididas de acordo com o tema, para facilitar a leitura e localização, e se encontram a seguir, na Tabela 3.2. O vocabulário, apresentado após as regras, complementa a descrição delas.

Tabela 3.2 – Regras da Base Três

| <b>Com relação à participação e membros</b>   |
|---|
| <b>Todo <u>membro efetivo</u> deve obrigatoriamente</b> estar matriculado no curso de Ciência da Computação <b>e</b> ter passado pelo <u>processo trainee</u> . |
| <b>O</b> mandato de um <u>membro efetivo</u> dura <b>6</b> meses.   |
| <b>É permitida</b> aos <u>membros efetivos</u> a candidatura à <u>Diretoria Executiva</u> .   |

|  |
|--|
| É <b>permitida</b> a participação de <u>membros efetivos</u> na Base Três por <b>no máximo 2</b> anos.   |
| <b>Com relação aos computadores da Base Três</b>   |
| É <b>proibido</b> realizar <u>trabalhos pessoais</u> nos computadores da Base Três.  |
| É <b>permitido</b> instalar programas nos computadores <b>apenas se</b> são necessários para realizar trabalho da empresa e é <b>necessário</b> que se comunique a instalação com os outros membros. |
| <b>Com relação a reclamações e problemas</b>   |
| <b>Todos os</b> problemas relacionados à empresa <b>devem obrigatoriamente</b> ser repassados à diretoria <b>ou</b> enviados por e-mail ao Departamento de Recursos Humanos.                         |
| <b>Com relação aos trajes</b>  |
| É <b>necessário</b> usar a <u>camisa da Base Três</u> dentro da empresa.   |
| É <b>proibido</b> usar <u>roupas inadequadas</u> dentro da empresa.  |
| É <b>obrigatório</b> usar a <u>camisa da Base Três</u> em <u>eventos externos</u> .  |
| <b>Com relação aos horários</b>  |
| É <b>obrigatório</b> que cada <u>membro efetivo</u> cumpra <b>pelo menos 6</b> horas semanais de trabalho.   |
| É <b>necessário</b> que a Base Três fique aberta durante o <u>horário de trabalho</u> .  |
| É <b>obrigatório</b> que faltas sejam comunicadas às pessoas do mesmo horário e justificadas ao Departamento de Recursos Humanos.  |
| <b>Com relação ao NEJ</b>  |
| É <b>necessário</b> que haja presença de <b>pelo menos um</b> membro de <b>cada</b> <u>departamento</u> nas reuniões dos comitês do NEJ.   |
| <b>Com relação à Tribuna de Minas</b>  |
| É <b>obrigatório</b> que cada <u>departamento</u> responda aos e-mails da Tribuna de Minas semanalmente, em <u>regime de rodízio</u> .   |
| É <b>possível</b> que o <u>departamento</u> da semana cobre a resposta de outros departamentos se ela não for de seu âmbito.   |
| <b>Com relação à limpeza</b>   |
| É <b>obrigatório</b> que cada <u>departamento</u> participe da <u>limpeza</u> , em regime de rodízio.  |
| <b>Com relação ao café</b>   |

|  |
|--|
| É <b>possível</b> que café seja feito <b>apenas se</b> a cafeteira seja lavada posteriormente. |
| É <b>obrigatório</b> que pingos de café que eventualmente ocorram sejam limpos imediatamente.  |
| <b>Regras Gerais</b>   |
| É <b>proibido</b> comer, <u>beber</u> ou fumar dentro da Base Três.                            |

O vocabulário que complementa a descrição das regras de negócio da Base Três encontra-se a seguir. Neste trabalho foram definidos apenas os nomes cujo entendimento pode não ser imediato.

#### <Vocabulário da Base Três>

Descrição: Conjunto de palavras utilizadas cotidianamente e relevantes para o trabalho.

Comunidade falante: Base Três.

Idioma: Português.

Nota: Apenas nomes cujo significado necessite esclarecimento foram incluídos.

#### Base Três

Definição: Também denominada EJI (Empresa Júnior de Informática) é uma associação civil sem fins lucrativos e com prazo de duração indeterminado, com foro e sede na cidade de Juiz de Fora, Estado de Minas Gerais.

#### membro efetivo

Definição: Estudante do Curso de Ciência da Computação que participa da Diretoria ou de um dos departamentos.

#### Diretoria Executiva

Definição: É composta de um Diretor Presidente, um Diretor Vice-Presidente, um Diretor do Departamento de Projetos, um Diretor do Departamento de Recursos Humanos, um Diretor do Departamento Administrativo-Financeiro e um Diretor do Departamento de Marketing e Relações Públicas.

#### processo trainee

Definição: Processo seletivo que objetiva a entrada e posterior integração de novos membros.

#### pessoas não-relacionadas ao trabalho

Definição: Qualquer um que não seja um membro efetivo da Base Três.

#### beber

Definição: Consumir líquidos que não sejam café ou água.

#### trabalhos pessoais

Definição: Trabalhos que não tem relação com as atividades desempenhadas na Base Três.

#### departamento(s)

Definição: Unidade organizacional da Base Três. Pode ser: Departamento de Marketing e Relações Públicas, Departamento de Recursos Humanos, Departamento Administrativo-Financeiro ou Departamento de Projetos.

#### camisa da Base Três

Definição: Camisa que possui o logotipo da Base Três e que permite a identificação de seus membros.

#### roupas inadequadas

Definição: Roupas que não condizem com um ambiente de uma empresa. Incluem: bermuda, boné, camiseta, óculos escuros e chinelos.

#### NEJ

Definição: Núcleo das Empresas Juniores (NEJ) é uma associação civil, sem fins lucrativos, que representa as empresas juniores da Universidade Federal de Juiz de Fora (UFJF) junto a órgãos públicos e privados, autoridades governamentais e sociedade em geral.

#### regime de rodízio

Definição: Alternância que ocorre entre os departamentos para a realização de tarefas.

#### limpeza

Definição: Zelo pelo ambiente da Base Três. É composto por: varrer o chão, tirar poeira e passar pano com produto de limpeza nas persianas, mesas, divisórias, armário, arquivo, vidros e computador.

#### eventos externos

Definição: Eventos que ocorrem fora da Base Três. São eles: Troca de gestão, reunião com cliente, NEJ e FEJEMG.

## FEJEMG

Definição: A Federação das Empresas Juniores do Estado de Minas Gerais (FEJEMG), associação civil com fins não econômicos e político-apartidária, é a entidade que representa oficialmente as Empresas juniores mineiras perante autoridades governamentais, clientes e sociedade em geral.

## horário de trabalho

Definição: Horário estipulado para o funcionamento da Base Três, das 9h às 19h.

### **3.4 Modelagem das Motivações**

Modelos de motivação de negócio foram abordados na Seção 2.6 e capturam o que a organização está tentando fazer (fins) e como ela faz (meios). Também procuram representar fatores que podem influenciar nessas escolhas (influenciadores).

Neste trabalho foi feita a modelagem das motivações da Base Três, na visão da gestão do segundo semestre de 2009. Nessa empresa, a cada semestre ocorre a troca de gestão, e cada presidência que assume define suas próprias aspirações para a gestão que vai ocorrer. Para realizar essa modelagem, foi utilizado BMM, definido na Seção 2.6.1. No intuito de melhorar a visualização, foi utilizada uma notação onde cada conceito do BMM é representado como uma caixa, e as caixas podem ser conectadas por setas, indicando relacionamento entre os conceitos. Essa notação não faz parte do padrão BMM, foi encontrada em BRG (2007) e é adotada em algumas ferramentas, como por exemplo o *Select Architect* da empresa *Select Business Solutions* (SELECT, 2009).

Na gestão do segundo semestre de 2009 foram definidos três fins, que são o coração do diagrama presente na Figura 3.2, divididos em duas metas e um objetivo. Metas não possuem data de cumprimento definido, ao passo que os objetivos possuem. No diagrama, os fins são representados em cinza escuro. Permeando os fins, estão os meios, representados em cinza claro. Os meios podem ser táticas ou estratégias. Estratégias podem ser descritas como táticas que possuem longa duração e que afetam uma grande parte da organização. Representados em branco estão o influenciador, sua avaliação, os riscos identificados na avaliação, e o que eles comprometem.

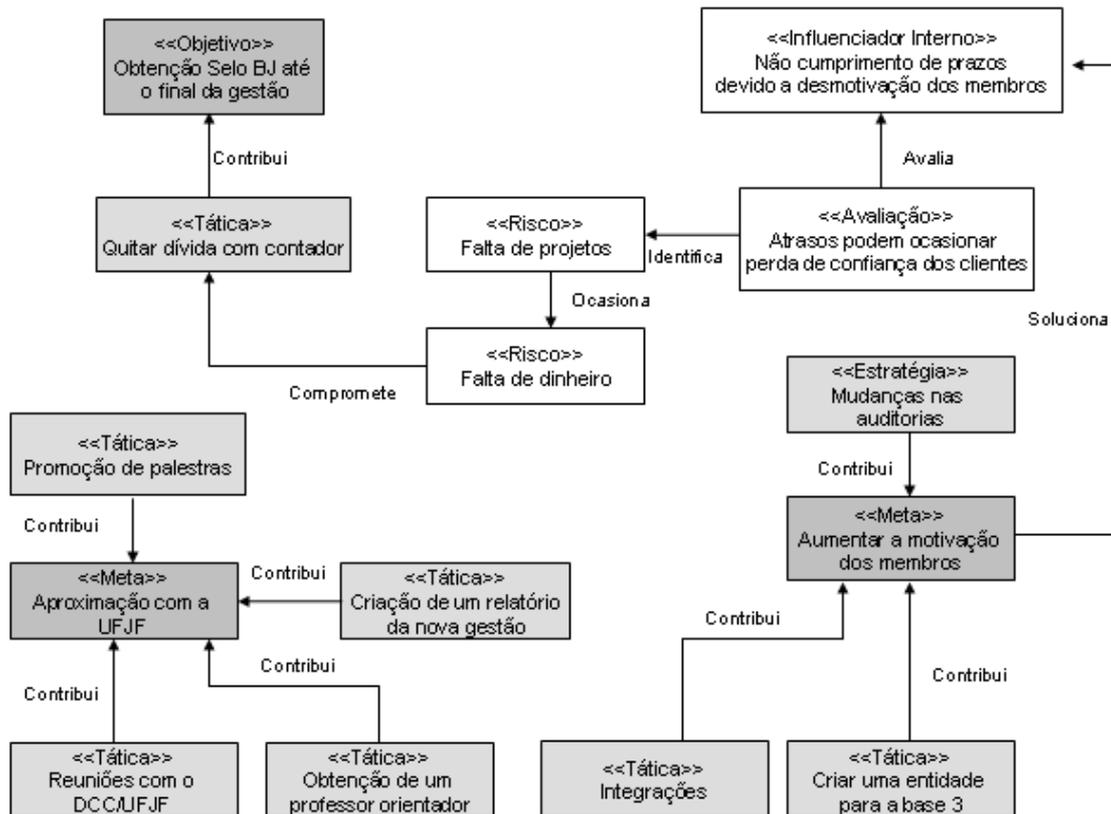


Figura 3.2 – BMM da gestão 2/2009 da Base Três

### 3.5 Modelagem dos Processos de Negócio

Processos de negócio foram descritos na Seção 2.2. Basicamente, um processo de negócio é uma coleção de tarefas executadas com o objetivo de se obter algum resultado para uma organização.

Para a modelagem dos processos da Base Três foi utilizado o padrão BPMN, descrito na Seção 2.2.1, e a ferramenta *BizAgi Process Modeler* que é gratuita e está disponível para *download* em BIZAGI (2009).

A Base Três possui muitos processos de negócio. Nesse trabalho são modelados três: o desenvolvimento de um projeto, o processo trainee e a troca de gestão. Eles serão abordados, respectivamente, nas seções: 3.51, 3.52 e 3.53.

#### 3.5.1 Processo de desenvolvimento de um projeto

Dentre as competências da Base Três está o desenvolvimento de sistemas de informação. Um projeto é uma seqüência de passos para a criação de um sistema

encomendado por um cliente. O diagrama de processo de negócio de desenvolvimento de projetos descreve desde a primeira reunião com o cliente até o *feedback* que acontece quinze dias após a entrega. É um processo que tem a participação de vários departamentos da empresa, por isso foi interessante a divisão do diagrama em raias, que permitem saber o responsável por determinada tarefa. Esse diagrama está presente na Figura 3.3 e na Figura 3.4.

### **3.5.2 Processo *Trainee***

Um dos objetivos da Base Três é preparar os alunos do Curso de Ciência da Computação da UFJF para o mercado de trabalho, e por isso, no começo de cada semestre acontece um processo seletivo para a entrada de novas pessoas na empresa. Dessa maneira os membros são sempre renovados, e um número maior de pessoas tem possibilidade de aprender.

O diagrama que representa esse processo está presente na Figura 3.5. Esse diagrama não possui raias, já que ele é essencialmente executado pelo Departamento de Recursos Humanos. Quando a participação de outros departamentos é relevante em determinada tarefa, isso é descrito por meio de anotações na mesma.

### **3.5.3 Processo de Troca de Gestão**

A cada seis meses acontece o processo de troca de gestão na Base Três, no qual membros efetivos podem se candidatar a posições de diretoria e os *trainees* podem ser efetivados. É composto por dois processos principais, as eleições, onde os candidatos apresentam seus planos e são votados, e a cerimônia que concretiza a troca de gestão.

O diagrama que representa o processo de eleição está presente na Figura 3.6, já o diagrama que representa a cerimônia de troca de gestão está na Figura 3.7.

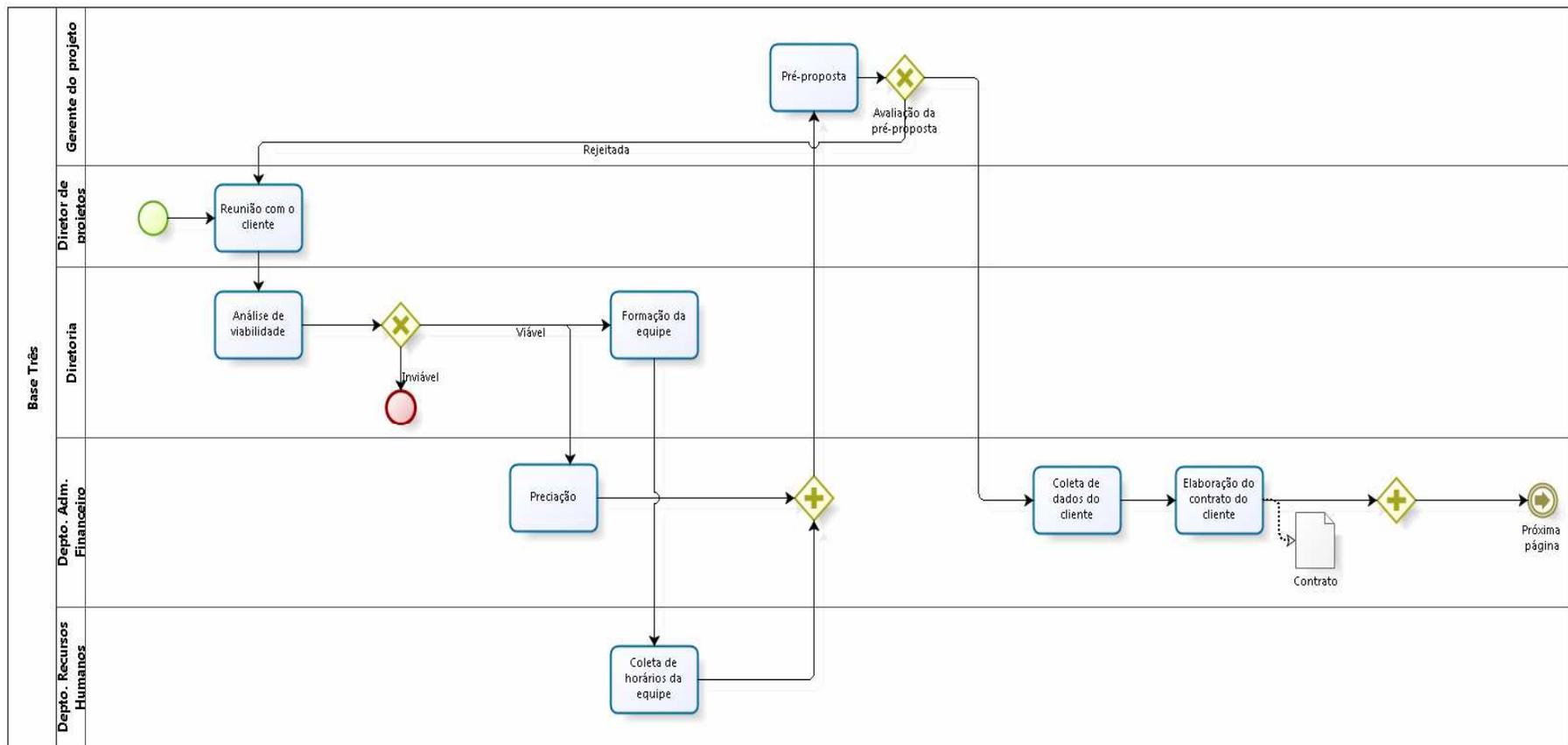


Figura 3.3 – Primeira parte do BPD de desenvolvimento de projetos

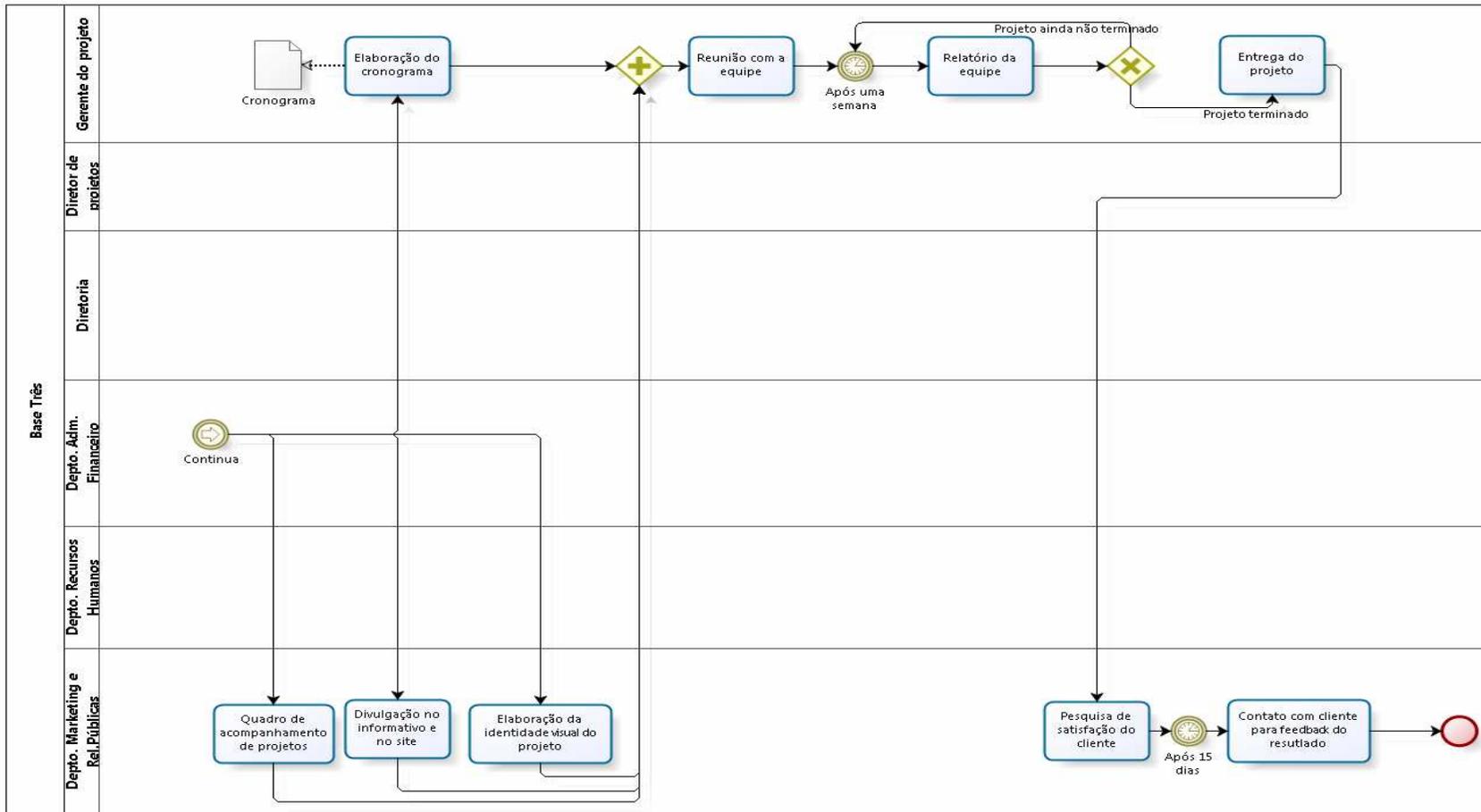


Figura 3.4 – Segunda parte do BPD de desenvolvimento de projetos

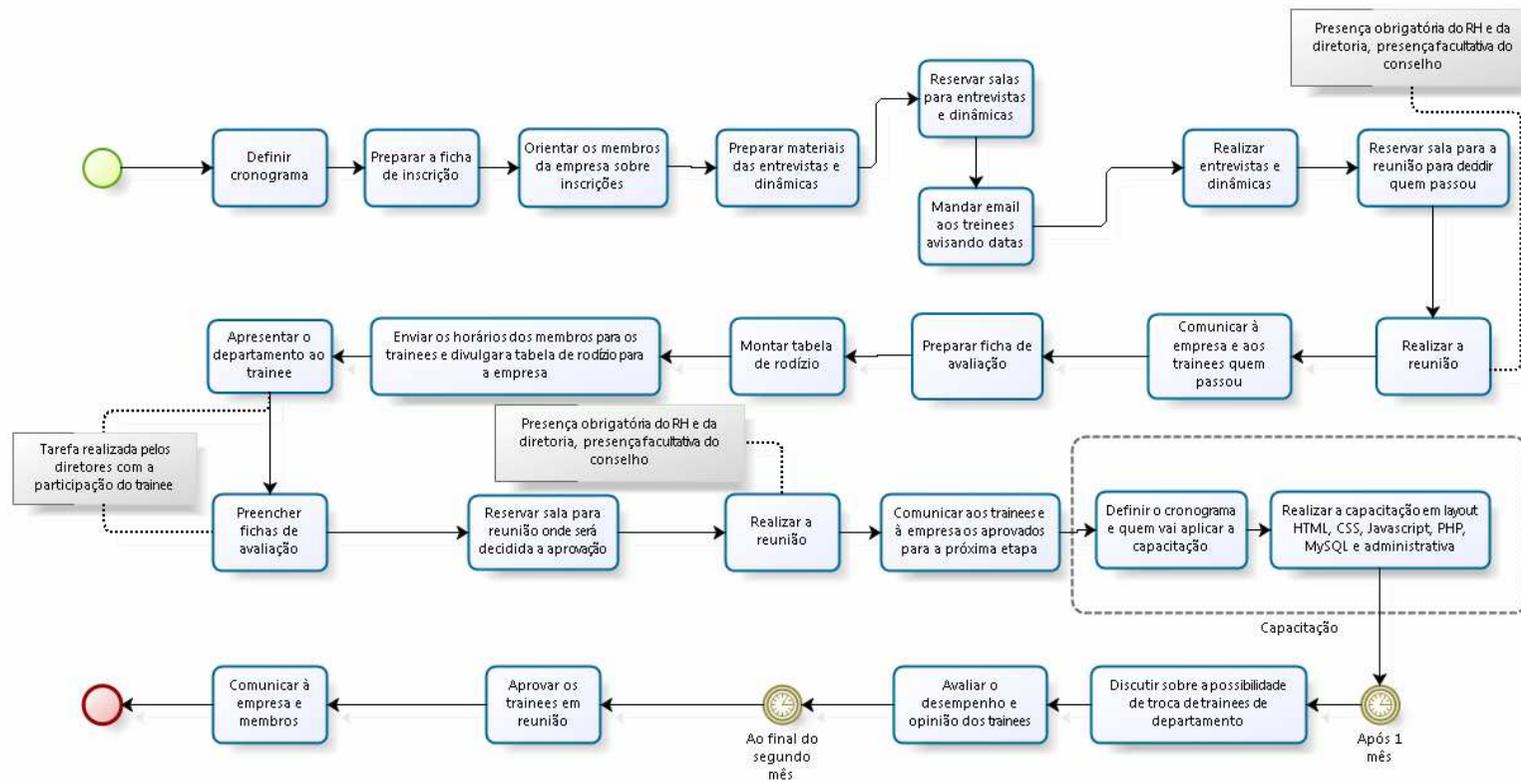


Figura 3.5 - BPD do Processo Trainee

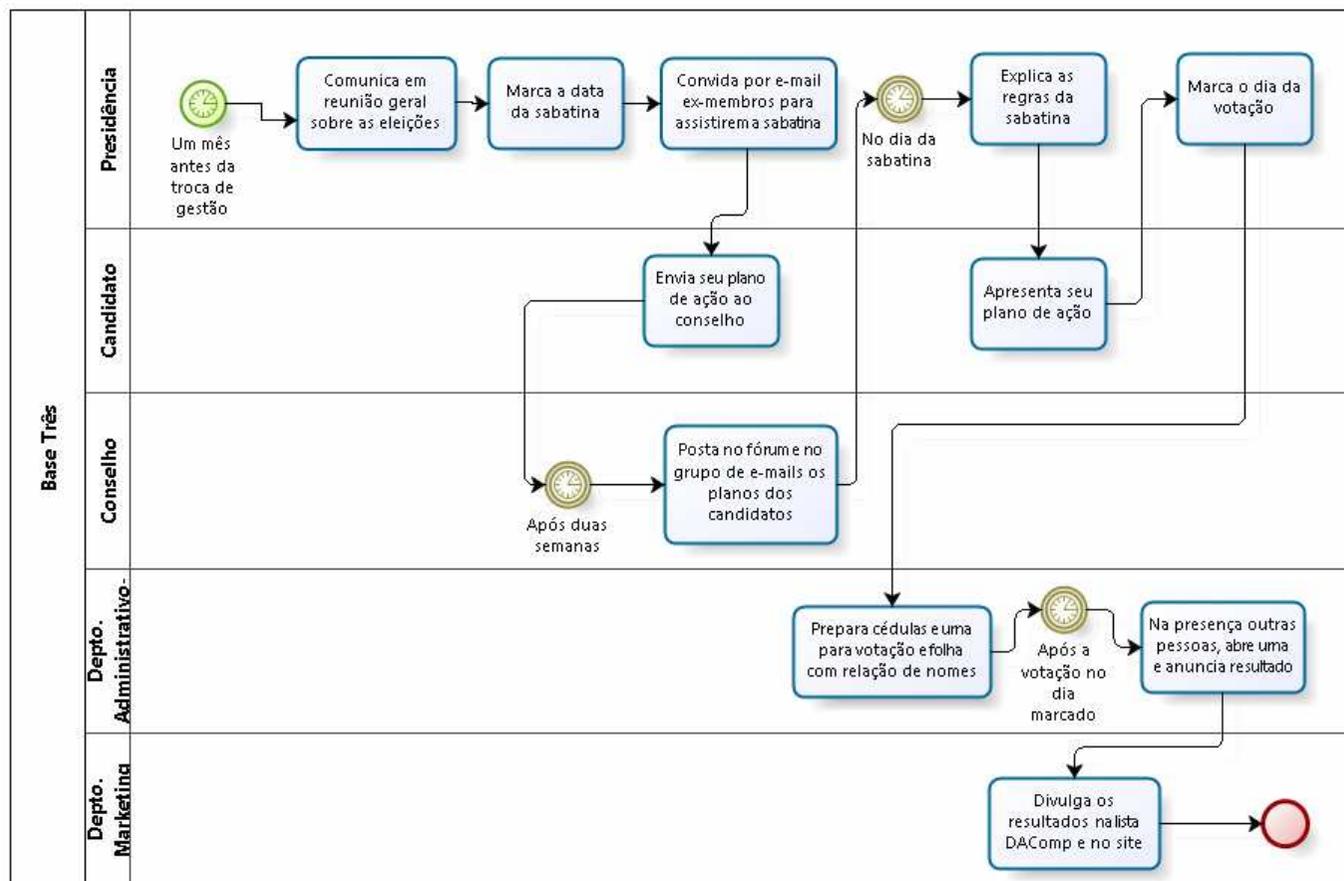


Figura 3.6 - BPD do processo de eleições

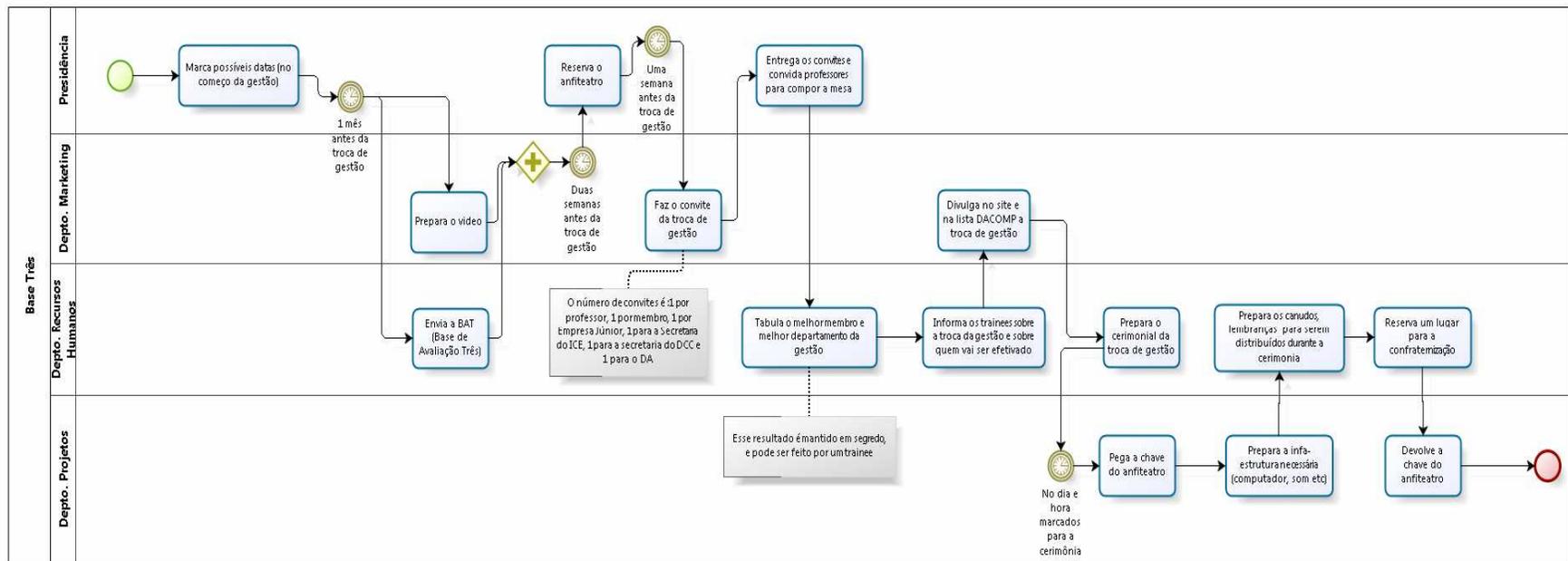


Figura 3.7 - BPD do processo de cerimônia da troca de gestão

### 3.6 Análise

Os modelos construídos foram validados e corrigidos para refletir a realidade. Apesar de estarem suficientemente fiéis, a sua construção encontrou várias dificuldades. Entre elas, vale citar a inexperiência do modelador, já que as linguagens que descrevem os modelos foram aprendidas com esse trabalho. Sabe-se que a modelagem é uma atividade que requer prática e a falta de experiência pode ter prejudicado os modelos construídos. Outra dificuldade importante foi a ausência de ferramentas. Dentre as linguagens usadas para construção de modelos, apenas para o BPMN foi encontrada ferramenta gratuita e de fácil utilização. As outras, ou não foram encontradas ferramentas (*Relationship Map*, SBVR), ou foram encontradas ferramentas que eram pagas (BMM). Para as linguagens sem ferramenta gratuita, utilizou-se o programa *PowerPoint*. Outro fator complicador foi o pouco tempo disponível para realizar a modelagem.

Com relação à empresa modelada - a Base Três, a principal dificuldade encontrada foi a ausência de documentação que pudesse servir de base para a construção dos modelos. Os documentos existentes estavam desatualizados, o que levou à construção de diagramas que não refletiam, inicialmente, a realidade atual. Em muitos casos, a documentação era informal, constituída por rascunhos manuscritos em folhas avulsas e anotações em agenda pessoal.

Porém, a modelagem da Base Três foi bastante facilitada devido ao interesse demonstrado pelo diretor-presidente, e a colaboração do mesmo na descrição da empresa, fornecimento de documentos e validação dos modelos construídos. Vale observar que a validação foi muito facilitada pelo fato da Base Três ser constituída por alunos do curso de Ciência da Computação, que já conhecem outros modelos e, assim, conseguiram compreender os novos modelos utilizados com facilidade.

Durante o processo de modelagem, observou-se que vários documentos descritivos da Base Três e seu negócio encontravam-se em locais de difícil acesso para os próprios membros da empresa, além de alguns deles estarem desatualizados. Isso parece evidenciar que tais documentos são pouco utilizados. A Base Três é uma empresa cujos membros são renovados a cada semestre, e é possível que algum conhecimento seja perdido a cada troca de gestão, devido a essa documentação deficiente. Espera-se que os modelos construídos neste trabalho possam ser usados no

futuro, como forma de preservação do conhecimento, melhoria na comunicação entre os membros, e como ferramenta de apoio em treinamentos.

### **3.7 Considerações Finais**

Esse capítulo buscou consolidar o estudo sobre as linguagens descritas no Capítulo 2, construindo modelos de uma situação existente no mundo real. Apesar da inexperiência do modelador e de várias outras dificuldades descritas na Seção 3.6, os modelos foram validados e refletem a realidade da Base Três. Além do conhecimento obtido, espera-se que os modelos construídos sejam úteis para a empresa, que possuía pouca documentação sobre o seu negócio.

## 4. Conclusão

Esse trabalho apresentou um apanhado geral sobre o que existe com relação a modelagem de negócio, descrevendo algumas linguagens com exemplos. Apesar da descrição de cada linguagem ter sido breve, acredita-se que foi suficiente para, ao menos, introduzir o leitor a cada uma delas, facilitando estudos futuros.

Com relação ao exemplo realizado com a modelagem da empresa júnior de informática da UFJF, conclui-se que o resultado foi satisfatório já que os modelos puderam ser construídos e foram validados por membros da empresa. Esses modelos também poderão ser úteis para a empresa no futuro.

A principal limitação desse trabalho foi o fato do estudo realizado não ter sido exaustivo, devido ao tempo curto disponível para a sua conclusão. O trabalho foi construído baseado na literatura, principalmente em BRIDGELAND e ZAHAVI (2009) e HAVEY (2005), e as linguagens ali descritas foram pesquisadas com mais detalhes em outras fontes. Muitas das linguagens estudadas são recentes, BMM e SBVR, por exemplo, com seus padrões publicados em 2008. Daí, a pouca literatura disponível sobre elas, fazendo com que a única fonte de consulta fosse, quase sempre, a própria especificação formal do padrão, o que dificultou a realização do trabalho.

Dos trabalhos encontrados durante a pesquisa para a construção deste, o mais relacionado é o de KO *et al* (2009), no qual os autores fazem um estudo exaustivo sobre gerência de processos de negócio, comparando as linguagens utilizadas para tal e identificando os pontos positivos e negativos de cada uma delas. Os autores mencionam não ser do conhecimento deles a existência de outro trabalho apresentando a mesma visão geral que eles oferecem. Como o trabalho de KO *et al.* é bastante recente (2009), acredita-se que esta monografia ajuda a cobrir uma falta de literatura desse tipo.

Como sugestão para trabalhos futuros, pode-se citar a modelagem da Base Três em outro conjunto de linguagens dentre aquelas descritas nesse trabalho e que não foram utilizadas, e posteriormente a realização de um estudo comparativo para concluir qual delas foi a melhor, segundo critérios a serem escolhidos.

## Referências

- AALST, Wil van der; HOFSTEDE, Arthur ter; KIEPUSZEWSKI, Bartosz *et al.* **Workflow Patterns**. Netherlands: Distributed And Parallel Databases, 2003. Disponível em: <<http://www.workflowpatterns.com/documentation/documents/wfs-pat-2002.pdf>>. Acesso em: 27 out. 2009.
- AALST, Wil van der; HOFSTEDE, Arthur ter. **YAWL: Yet Another Workflow Language (Revised version)**. Brisbane: Queensland University Of Technology, 2003. Disponível em: <<http://www.yawlfoundation.org/documents/yawlrevtech.pdf>>. Acesso em: 27 out. 2009.
- ACTANO (Munique). **Glossary**. Disponível em: <[http://www.actano.de/20911\\_EN-What%20s\\_new-Glossary.htm](http://www.actano.de/20911_EN-What%20s_new-Glossary.htm)>. Acesso em: 13 out. 2009.
- BASE TRÊS (Juiz de Fora). **Base Três Consultoria em Informática**. Disponível em: <<http://www.basetres.ufjf.br>>. Acesso em: 30 out. 2009.
- BIZAGI. **BizAgi Process Modeler**. Disponível em: <[http://www.bizagi.com/index.php?option=com\\_content&view=article&id=95&Itemid=107](http://www.bizagi.com/index.php?option=com_content&view=article&id=95&Itemid=107)>. Acesso em: 1 dez. 2009.
- BPMI, **Business Process Management Initiative**. Disponível em: <<http://www.bpmi.org/>>. Acesso em: 1 dez. 2009.
- BRG. **Defining Business Rules: What Are They Really?** Seattle, 2000. Disponível em: <[http://www.businessrulesgroup.org/first\\_paper/BRG-whatIsBR\\_3ed.pdf](http://www.businessrulesgroup.org/first_paper/BRG-whatIsBR_3ed.pdf)>. Acesso em: 10 nov. 2009.
- BRG. **The Business Motivation Model: Business Governance in a Volatile World**. Seattle, 2007. Disponível em: <[http://www.businessrulesgroup.org/second\\_paper/BRG-BMM.pdf](http://www.businessrulesgroup.org/second_paper/BRG-BMM.pdf)>. Acesso em: 1 dez. 2009.
- BRIDGELAND, David; ZAHAVI, Ron. **Business Modeling: A Practical Guide to Realizing Business Value**. Burlington: Morgan Kaufman, 2009.
- HARRINGTON, James. **Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness**. New York: Mcgraw-hill, 1991.

- HAVEY, Mike. **Essential Business Process Modeling**. O'Reilly Media, 2005.
- HOLLINGSWORTH, David. **Workflow Management Coalition - The Workflow Reference Model**. Winchester. 1995. Disponível em: <<http://www.wfmc.org/standards/docs/tc003v11.pdf>>. Acesso em: 27 out. 2009.
- JOHANSSON, Henry *et al.* **Business Process Reengineering: Break Point Strategies for Market Dominance**. Chichester: John Wiley & Sons Ltd, 1993.
- JURIC, Matjaz B.; MATHEW, Benny K.; SARANG, Poornachandra. **Business Process Execution Language for Web Services**. 2. ed. Birmingham: Packet Publishing, 2006.
- KO, Ryan K. L.; LEE, Stephen S G; LEE, Eng Wah. Business Process Management Standards: A survey. **Business Process Management Journal**, p. 744-791. 2009. Disponível em: <[http://ryanko.files.wordpress.com/2008/12/bpmj\\_stds\\_survey\\_koleelee.pdf](http://ryanko.files.wordpress.com/2008/12/bpmj_stds_survey_koleelee.pdf)>. Acesso em: 20 out. 2009.
- KORHERR, Birgit; LIST, Beate. AUML 2 Profile for Event Driven Process Chains. **Research And Practical Issues Of Enterprise Information Systems**, Viena, p. 161-172. 24 abr. 2006. Disponível em: <[http://publik.tuwien.ac.at/files/pub-inf\\_4133.pdf](http://publik.tuwien.ac.at/files/pub-inf_4133.pdf)>. Acesso em: 20 out. 2009.
- LEYMANN, Frank. **Web Services Flow Language (WSFL 1.0)**. Disponível em: <<http://xml.coverpages.org/WSFL-Guide-200110.pdf>>. Acesso em: 27 out. 2009.
- MAXIMIANO, Antonio Cesar Amaru. **Introdução à Administração**. 5. ed. São Paulo: Atlas, 2000.
- MICROSOFT. **XLANG/s Language**. Disponível em: <[http://msdn.microsoft.com/en-us/library/aa577463\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/aa577463(BTS.10).aspx)>. Acesso em: 27 out. 2009.
- MILES, Russell; HAMILTON, Kim. **Learning UML 2.0**. Sebastopol: O'reilly Media, 2006.
- OASIS. **OASIS Standards**. Disponível em: <<http://www.oasis-open.org/specs/index.php>>. Acesso em: 27 out. 2009.

- OMG, **Business Process Modeling Notation**, V1.0, 2006. Disponível em: <<http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>>. Acesso em: 13 out. 2009.
- OMG, **Business Process Modeling Notation**, V1.1, 2008a. Disponível em: <<http://www.omg.org/spec/BPMN/1.1/PDF/>>. Acesso em: 13 out. 2009.
- OMG, **Semantics of Business Vocabulary and Business Rules**, V1.0, 2008b. Disponível em: < <http://www.omg.org/spec/SBVR/1.0/PDF/>> Acesso em: 3 nov. 2009.
- OMG, **Business Motivation Model**, V1.0, 2008c. Disponível em: < <http://www.omg.org/spec/SBVR/1.0/PDF/>> Acesso em: 3 nov. 2009.
- RULESPEAK. **Home Page**. Disponível em: <<http://www.rulespeak.com/en/>>. Acesso em: 03 nov. 2009.
- RUMMLER, Geary A; BRACHE, Alan P. **Improving Performance**: How to manage the white space on the organizational chart. San Francisco: Jossey-Bass, 1995.
- SAMPAIO, Marcus Costa. **Diagrama de Atividades**. Disponível em: <[http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/atividades/diag\\_atividades.htm](http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/atividades/diag_atividades.htm)>. Acesso em: 20 out. 2009.
- SELECT. **Select Architect (BMM, BPMN, UML)**. Disponível em: <<http://www.selectbs.com/adt/analysis-and-design/select-architect>>. Acesso em: 1 dez. 2009.
- SUN. **JSR 207: Process Definition for Java**. Disponível em: <<http://jcp.org/en/jsr/detail?id=207>>. Acesso em: 27 out. 2009.
- WESKE, Mathias. **EPC and workflow nets**. Disponível em: <<http://www.bpm-book.com/BPMbook/Exercise4-31>>. Acesso em: 20 out. 2009.
- WFMC. **Home Page**. Disponível em: <<http://www.wfmc.org/>>. Acesso em: 27 out. 2009.
- WHITE, Stephen. **Mapping BPMN to BPEL Example**. Estados Unidos: IBM Corp, 2005. Disponível em: <<http://www.bpmn.org/Documents/Mapping%20BPMN%20to%20BPEL%20Example.pdf>>. Acesso em: 10 nov. 2009.

## Anexo 1: Exemplo de uso de BPEL

O exemplo a seguir é escrito em BPEL e descreve um processo de requisição de seguro. Os comentários escritos entre `<!--` e `-->` esclarecem o que cada trecho de código quer dizer.

```
1  <process name="InsuranceClaim"
2    targetNamespace="http://acm.org/samples"
3    suppressJoinFailure="yes"
4    xmlns:tns="http://acm.org/samples"
5    xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
6    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
7    xmlns:addressing="http://schemas.xmlsoap.org/ws/2003/03/addressing"
8    xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
9
10   <!--
11     Parceiros no processo:
12     cliente - aplicação que inicia e mata o processo
13     worklist - serviço que gerencia as atividades manuais
14   -->
15   <partnerLinks>
16
17     <partnerLink name="client" partnerLinkType="tns:InsuranceClaim"
18       myRole="InsuranceClaimProvider" />
19     <partnerLink name="worklist" partnerLinkType="task:TaskManager"
20       partnerRole="TaskManager" myRole="TaskManagerRequester" />
21   </partnerLinks>
22
23   <!-- Variáveis nível do processo -->
24   <variables>
25     <variable name="status" type="xsd:string" />
26     <variable name="initiateMsg" messageType="tns:InsuranceClaimMsg" />
27     <variable name="killEv" messageType="tns:InsuranceClaimMsg" />
28     <variable name="taskResponse" messageType="task:taskMessage" />
29   </variables>
30
31   <!-- Relacionamento para mensagens serão feitos usando o campo ClaimID -->
32   <correlationSets>
33     <correlationSet name="claim" properties="tns:claimID" />
34   </correlationSets>
35
36   <!-- Captura erros e os conserta de forma manual -->
37   <faultHandlers>
38     <catchAll>
39       <empty name="PlaceholderForManualFix" />
40     </catchAll>
41   </faultHandlers>
42
43   <!-- Recebe globalmente um evento de morte (relacionado com o ClaimID do
44     processo que o iniciou originalmente) e termina o processo. -->
45   <eventHandlers>
46     <onMessage partnerLink="client" portType="tns:InsuranceClaim"
47       operation="kill" variable="killEv">
48       <correlations>
49         <correlation set="claim" initiate="no" />
50       </correlations>
51       <sequence>
52         <empty /><!--Aqui algo pode ser feito, como notificar alguém da morte-->
53         <terminate name="killClaim" />
54       </sequence>
55     </onMessage>
56   </eventHandlers>
57
58   <sequence>
59
60     <!--Começa-se com uma atividade de recebimento que recebe a mensagem inicial.
61       relaciona-se com o claim definido anteriormente
62     -->
63     <receive partnerLink="client" portType="tns:InsuranceClaim"
```

```

64         operation="initiate" variable="initiateMsg" createInstance="yes"
65         name="initiateEvent">
66         <correlations>
67             <correlation set="claim" initiate="yes"/>
68         </correlations>
69     </receive>
70
71     <!-- Invoca o parceiro worklist para avaliar a mensagem recebida -->
72     <invoke name="evalClaim" partnerLink="worklist" portType="task:TaskManager"
73         operation="evalClaim" inputVariable="initiateMsg"/>
74
75     <!-- Recebe a resposta ou um timeout. -->
76     <pick name="analyzePick">
77         <onMessage partnerLink="worklist" portType="task:TaskManagerCallback"
78             operation="onTaskResult" variable="taskResponse">
79             <!-- Extrai o status da resposta e o atribui à variável 'status'-->
80             <assign name="setStatus">
81                 <copy>
82                     <from variable="taskResponse" part="payload"
83                         query="/tns:taskMessage/tns:result="/>
84                     <to variable="status"/>
85                 </copy>
86             </assign>
87         </onMessage>
88         <!-- Timeout! 2 dias se passaram. Avança para o próximo passo -->
89         <onAlarm for="PT2D">
90             <sequence>
91                 <!-- Chama o service parceiro para prosseguir -->
92                 <invoke name="evalClaim" partnerLink="worklist"
93                     portType="task:TaskManager" operation="escalateClaim"
94                     inputVariable="initiateMsg"/>
95                 <!-- Recebe a resposta -->
96                 <receive name="receiveTaskResult" partnerLink="worklist"
97                     portType="task:TaskManagerCallback"
98                     operation="onTaskResult" variable="taskResponse"/>
99                 <!-- Extrai o status da resposta e o atribui à variável 'status'-->
100                <assign name="setStatus">
101                    <copy>
102                        <from variable="taskResponse" part="payload"
103                            query="/tns:taskMessage/tns:result="/>
104                        <to variable="status"/>
105                    </copy>
106                </assign>
107            </sequence>
108        </onAlarm>
109    </pick>
110
111    <!-- Age de acordo com o status do processo: 'rejected' e 'accepted'
112         são corretos. Qualquer outro, lança uma exceção (fault) -->
113    <switch name="resultEval">
114        <case condition="bpws:getVariableData('status')='rejected'">
115            <empty> <!-- Realiza ações correspondentes à requisição rejeitada -->
116        </case>
117        <case condition="bpws:getVariableData('status')='accepted'">
118            <empty> <!-- Realiza ações correspondentes à requisição aceita -->
119        </case>
120        <otherwise>
121            <throw name="illegalStatus" faultName="illegalStatus"/>
122        </otherwise>
123    </switch>
124 </sequence>
125 </process>

```

A lógica para esse processo é bem simples: quando uma requisição chega, um agente da seguradora a avalia e determina se ela será aceita ou não. Caso o agente não responda em um período de dois dias, a atividade é enviada a um gerente toma uma decisão acelerada do pedido, aceitando ou rejeitando. Esse processo pode ser finalizado a qualquer momento. O diagrama de atividade presente na Figura 2.4 representa o processo.

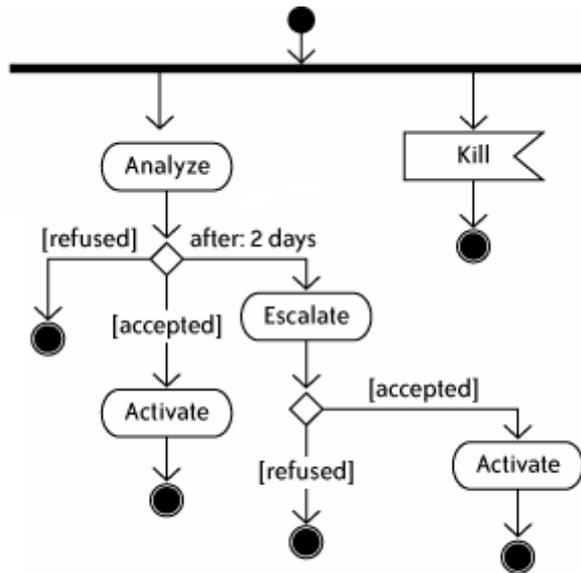


Figura A1.1 - Um processo de requisição de seguro

O código que implementa essa lógica é um documento XML, cujo elemento raiz é o *process*. Esse elemento possui uma propriedade que nomeia o processo, no caso *InsuranceClaim* (linha 1). O coração do processo é a atividade do tipo *sequence* presente nas linhas 54-124, que contém um conjunto de atividades filhas que executam em seqüência. As atividades filhas nesse exemplo são:

- *Receive* (linhas 63-69): Tem o propósito de escutar mensagens que representam uma requisição de seguro. O atributo *createInstance="yes"* na linha 64 significa que o *receive* é o gatilho que inicia o processo.
- *Invoke* (linhas 72-73): Chama uma operação oferecida por um parceiro (no caso, a operação *evalClaim* do parceiro *worklist*). O parceiro escolhe um agente da seguradora para avaliar o pedido. Vários dias podem se passar até que o agente faça a decisão.
- *Pick* (linhas 76-109): Essa atividade aguarda por um dos dois eventos que podem ocorrer, a resposta chegar do *worklist* (gerenciado pelo *onMessage* das linhas 77-87) ou um *timeout* (tratado pelo *onAlarm* nas linhas 89-108). No primeiro caso, a *worklist* chama o processo de volta usando através do *onTaskResult* (linha 78). O *assign* presente nas linhas 80-86 copia o valor do resultado da mensagem oriunda do *worklist* e copia para a variável *status* (declarada na linha 25). O *timeout* de dois dias é definido pela condição *for="PT2D"* na linha 89. Esse *timeout* ativa chama a operação

*escalateClaim* do parceiro *worklist* (linhas 92-94). O *receive* existente nas linhas 96-98 espera pelo resultado, e o *assign* das linhas 100-106 copia o resultado para a variável *status*.

- *Switch* (linhas 113-123): É uma estrutura do tipo *ou exclusivo* que aceita (caso tratado nas linhas 114-116) ou rejeita (caso tratado nas linhas 120-122) baseado no valor da variável *status*. A maneira como o processo trata cada caso não está representada no exemplo. Caso o valor da variável *status* não seja nenhum dos dois casos citados, a atividade lança um *fault* chamado *illegalStatus*, que ativa o *faultHandler* presente nas linhas 37-42.

O pedido pode ser finalizado a qualquer momento, e isso é representado no XML pelo *eventHandlers* das linhas 45-56; se isso ocorrer, o processo finaliza utilizando o *terminate* presente na linha 53.

## Anexo 2: Exemplo de código XPDL

O código a seguir foi gerado pela ferramenta JAWE (Enhydra Java Workflow Editor), ilustrada na figura 2.8.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Package Id="Travel" Name="Travel" xmlns=http://www.wfmc.org/2002/XPDL1.0
3   xmlns:xpdl=http://www.wfmc.org/2002/XPDL1.0
4   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
5   xsi:schemaLocation="http://www.wfmc.org/2002/XPDL1.0
6   http://wfmc.org/standards/docs/TC-1025_schema_10_xpdl.xsd">
7   <PackageHeader>
8     <XPDLVersion>1.0</XPDLVersion>
9     <Vendor>Together</Vendor>
10    <Created>2004-09-20 22:13:58</Created>
11  </PackageHeader>
12  <RedefinableHeader PublicationStatus="UNDER_TEST" />
13  <ConformanceClass GraphConformance="NON_BLOCKED" />
14
15  <!-- Três aplicações são usadas pelo processo: hotel, air e car, cada uma
16   delas recebe um itinerário como entrada e retorna um resultado -->
17  <Applications>
18    <Application Id="HotelApp" Name="HotelApp">
19      <FormalParameters>
20        <FormalParameter Id="itinerary" Mode="IN">
21          <DataType>
22            <BasicType Type="STRING" />
23          </DataType>
24        </FormalParameter>
25        <FormalParameter Id="result" Mode="OUT">
26          <DataType>
27            <BasicType Type="STRING" />
28          </DataType>
29        </FormalParameter>
30      </FormalParameters>
31    </Application>
32    <Application Id="AirApp" Name="AirApp">
33      <FormalParameters>
34        <FormalParameter Id="itinerary" Mode="IN">
35          <DataType>
36            <BasicType Type="STRING" />
37          </DataType>
38        </FormalParameter>
39        <FormalParameter Id="result" Mode="OUT">
40          <DataType>
41            <BasicType Type="STRING" />
42          </DataType>
43        </FormalParameter>
44      </FormalParameters>
45    </Application>
46    <Application Id="CarApp" Name="CarApp">
47      <FormalParameters>
48        <FormalParameter Id="itinerary" Mode="IN">
49          <DataType>
50            <BasicType Type="STRING" />
51          </DataType>
52        </FormalParameter>
53        <FormalParameter Id="result" Mode="OUT">
54          <DataType>
55            <BasicType Type="STRING" />
56          </DataType>
57        </FormalParameter>
58      </FormalParameters>
59    </Application>
60  </Applications>
61
62  <!-- O processo recebe o itinerário como parâmetro.
63   Ele usa três campos de dados do tipo string, representando
64   os resultados das chamadas feitas às aplicações. -->
65  <WorkflowProcesses>
66    <WorkflowProcess AccessLevel="PUBLIC" Id="TravelWorkflow"
67      Name="TravelWorkflow">
```

```

68 <ProcessHeader DurationUnit="D">
69   <Created>2004-09-20 22:15:14</Created>
70 </ProcessHeader>
71 <RedefinableHeader PublicationStatus="UNDER_TEST" />
72 <FormalParameters>
73   <FormalParameter Id="itinerary" Mode="IN">
74     <DataType>
75       <BasicType Type="STRING" />
76     </DataType>
77   </FormalParameter>
78 </FormalParameters>
79 <DataFields>
80   <DataField Id="airResult" IsArray="FALSE">
81     <DataType>
82       <BasicType Type="STRING" />
83     </DataType>
84   </DataField>
85   <DataField Id="carResult" IsArray="FALSE">
86     <DataType>
87       <BasicType Type="STRING" />
88     </DataType>
89   </DataField>
90   <DataField Id="hotelResult" IsArray="FALSE">
91     <DataType>
92       <BasicType Type="STRING" />
93     </DataType>
94   </DataField>
95 </DataFields>
96 <Participants>
97   <Participant Id="TravelAgent">
98     <ParticipantType Type="ROLE" />
99   </Participant>
100 </Participants>
101
102 <Activities>
103
104   <!-- A primeira atividade é GetItinerary. É uma atividade manual
105     realizada por um agente de viagens. Quando ela é completada
106     ocorre uma divisão em três atividades paralelas. -->
107   <Activity Id="GetItinerary" Name="GetItinerary">
108     <Implementation>
109       <No/>
110     </Implementation>
111     <Performer>TravelAgent</Performer>
112     <StartMode>
113       <Manual/>
114     </StartMode>
115     <FinishMode>
116       <Manual/>
117     </FinishMode>
118     <TransitionRestrictions>
119       <TransitionRestriction>
120         <Split Type="AND">
121           <TransitionRefs>
122             <TransitionRef Id="TravelWorkflow_Tra3" />
123             <TransitionRef Id="TravelWorkflow_Tra4" />
124             <TransitionRef Id="TravelWorkflow_Tra1" />
125           </TransitionRefs>
126         </Split>
127       </TransitionRestriction>
128     </TransitionRestrictions>
129     <ExtendedAttributes>
130       <ExtendedAttribute Name="ParticipantID" Value="Agency" />
131       <ExtendedAttribute Name="XOffset" Value="160" />
132       <ExtendedAttribute Name="YOffset" Value="60" />
133     </ExtendedAttributes>
134   </Activity>
135
136   <!-- SubmitAir chama a aplicação air, passando um itinerário
137     e recebendo um resultado. -->
138   <Activity Id="SubmitAir" Name="SubmitAir">
139     <Implementation>
140       <Tool Id="AirApp" Type="APPLICATION">
141         <ActualParameters>
142           <ActualParameter>itinerary</ActualParameter>
143           <ActualParameter>airResult</ActualParameter>
144         </ActualParameters>

```

```

145         </Tool>
146     </Implementation>
147     <Performer>TravelAgent</Performer>
148     <StartMode>
149         <Automatic/>
150     </StartMode>
151     <FinishMode>
152         <Automatic/>
153     </FinishMode>
154     <ExtendedAttributes>
155         <ExtendedAttribute Name="ParticipantID" Value="Agency"/>
156         <ExtendedAttribute Name="XOffset" Value="270"/>
157         <ExtendedAttribute Name="YOffset" Value="10"/>
158     </ExtendedAttributes>
159 </Activity>
160
161 <!-- SubmitHotel chama a aplicação hotel, passando um itinerário
162 e recebendo um resultado. -->
163 <Activity Id="SubmitHotel" Name="SubmitHotel">
164     <Implementation>
165         <Tool Id="HotelApp" Type="APPLICATION">
166             <ActualParameters>
167                 <ActualParameter>itinerary</ActualParameter>
168                 <ActualParameter>hotelResult</ActualParameter>
169             </ActualParameters>
170         </Tool>
171     </Implementation>
172     <Performer>TravelAgent</Performer>
173     <StartMode>
174         <Automatic/>
175     </StartMode>
176     <FinishMode>
177         <Automatic/>
178     </FinishMode>
179     <ExtendedAttributes>
180         <ExtendedAttribute Name="ParticipantID" Value="Agency"/>
181         <ExtendedAttribute Name="XOffset" Value="270"/>
182         <ExtendedAttribute Name="YOffset" Value="60"/>
183     </ExtendedAttributes>
184 </Activity>
185
186 <!-- SubmitCar chama a aplicação car, passando um itinerário
187 e recebendo um resultado. -->
188 <Activity Id="SubmitCar" Name="SubmitCar">
189     <Implementation>
190         <Tool Id="CarApp" Type="APPLICATION">
191             <ActualParameters>
192                 <ActualParameter>itinerary</ActualParameter>
193                 <ActualParameter>carResult</ActualParameter>
194             </ActualParameters>
195         </Tool>
196     </Implementation>
197     <Performer>TravelAgent</Performer>
198     <StartMode>
199         <Automatic/>
200     </StartMode>
201     <FinishMode>
202         <Automatic/>
203     </FinishMode>
204     <ExtendedAttributes>
205         <ExtendedAttribute Name="ParticipantID" Value="Agency"/>
206         <ExtendedAttribute Name="XOffset" Value="270"/>
207         <ExtendedAttribute Name="YOffset" Value="120"/>
208     </ExtendedAttributes>
209 </Activity>
210
211 <!-- Atividade manual, realizada por um agente de viagem,
212 enviar o itinerário para o cliente. -->
213 <Activity Id="MailItinerary" Name="MailItineray">
214     <Implementation>
215         <No/>
216     </Implementation>
217     <Performer>TravelAgent</Performer>
218     <StartMode>
219         <Manual/>
220     </StartMode>
221     <FinishMode>

```

```

222         <Manual/>
223     </FinishMode>
224     <ExtendedAttributes>
225         <ExtendedAttribute Name="ParticipantID" Value="Agency"/>
226         <ExtendedAttribute Name="XOffset" Value="480"/>
227         <ExtendedAttribute Name="YOffset" Value="20"/>
228     </ExtendedAttributes>
229 </Activity>
230
231 <!-- Atividade manual, realizada por um agente de viagem, ligar
232      para o cliente pedindo para que faça mudanças no itinerário -->
233 <Activity Id="CallCustomerForMods" Name="CallCustomerForMods">
234     <Implementation>
235         <No/>
236     </Implementation>
237     <Performer>TravelAgent</Performer>
238     <StartMode>
239         <Manual/>
240     </StartMode>
241     <FinishMode>
242         <Manual/>
243     </FinishMode>
244     <ExtendedAttributes>
245         <ExtendedAttribute Name="ParticipantID" Value="Agency"/>
246         <ExtendedAttribute Name="XOffset" Value="480"/>
247         <ExtendedAttribute Name="YOffset" Value="100"/>
248     </ExtendedAttributes>
249 </Activity>
250
251 <!-- Atividade route: junta os 3 fluxos de entrada
252      e faz uma divisão do tipo ou exclusivo
253      para uma das duas atividades manuais -->
254 <Activity Id="TravelWorkflow_Act7" Name="Route">
255     <Route/>
256     <StartMode>
257         <Automatic/>
258     </StartMode>
259     <FinishMode>
260         <Automatic/>
261     </FinishMode>
262     <TransitionRestrictions>
263         <TransitionRestriction>
264             <Join Type="AND"/>
265             <Split Type="XOR">
266                 <TransitionRefs>
267                     <TransitionRef Id="TravelWorkflow_Tra12"/>
268                     <TransitionRef Id="TravelWorkflow_Tra11"/>
269                 </TransitionRefs>
270             </Split>
271         </TransitionRestriction>
272     </TransitionRestrictions>
273     <ExtendedAttributes>
274         <ExtendedAttribute Name="ParticipantID" Value="Agency"/>
275         <ExtendedAttribute Name="XOffset" Value="380"/>
276         <ExtendedAttribute Name="YOffset" Value="60"/>
277     </ExtendedAttributes>
278 </Activity>
279 </Activities>
280
281 <!-- Definição das transições:
282      GetItinerary direciona para cada uma das atividades submit.
283      Cada uma das atividades submit direciona para a atividade route.
284      A atividade route direciona para MailItinerary se
285      airResult="ok" e carResult="ok" e hotelResult="ok".
286      Caso contrário, direciona para GetMods. -->
287 <Transitions>
288
289     <Transition From="GetItinerary" Id="TravelWorkflow_Tra1"
290         Name="Transition" To="SubmitAir">
291         <ExtendedAttributes>
292             <ExtendedAttribute Name="RoutingType" Value="NOROUTING"/>
293         </ExtendedAttributes>
294     </Transition>
295
296     <Transition From="GetItinerary" Id="TravelWorkflow_Tra3"
297         Name="Transition" To="SubmitHotel">
298         <ExtendedAttributes>

```

```

299         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
300     </ExtendedAttributes>
301 </Transition>
302
303 <Transition From="GetItinerary" Id="TravelWorkflow_Tra4"
304     Name="Transition" To="SubmitCar">
305     <ExtendedAttributes>
306         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
307     </ExtendedAttributes>
308 </Transition>
309
310 <Transition From="SubmitAir" Id="TravelWorkflow_Tra8"
311     Name="Transition" To="TravelWorkflow_Act7">
312     <ExtendedAttributes>
313         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
314     </ExtendedAttributes>
315 </Transition>
316
317 <Transition From="SubmitCar" Id="TravelWorkflow_Tra9"
318     Name="Transition" To="TravelWorkflow_Act7">
319     <ExtendedAttributes>
320         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
321     </ExtendedAttributes>
322 </Transition>
323
324 <Transition From="SubmitHotel" Id="TravelWorkflow_Tra10"
325     Name="Transition" To="TravelWorkflow_Act7">
326     <ExtendedAttributes>
327         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
328     </ExtendedAttributes>
329 </Transition>
330
331 <Transition From="TravelWorkflow_Act7" Id="TravelWorkflow_Tra11"
332     Name="Transition" To="MailItinerary">
333     <Condition Type="CONDITION">
334         airResult="ok" and carResult="ok" and hotelResult="ok"
335     </Condition>
336     <ExtendedAttributes>
337         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
338     </ExtendedAttributes>
339 </Transition>
340
341 <Transition From="TravelWorkflow_Act7" Id="TravelWorkflow_Tra12"
342     Name="Transition" To="CallCustomerForMods">
343     <Condition Type="OTHERWISE" />
344     <ExtendedAttributes>
345         <ExtendedAttribute Name="RoutingType" Value="NOROUTING" />
346     </ExtendedAttributes>
347 </Transition>
348 </Transitions>
349
350 <ExtendedAttributes>
351     <ExtendedAttribute Name="StartOfWorkflow"
352         Value="Agency;GetItinerary;90;60;NOROUTING" />
353     <ExtendedAttribute Name="EndOfWorkflow"
354         Value="Agency;CallCustomerForMods;580;100;NOROUTING" />
355     <ExtendedAttribute Name="EndOfWorkflow"
356         Value="Agency;MailItinerary;580;20;NOROUTING" />
357     <ExtendedAttribute Name="ParticipantVisualOrder"
358         Value="Agency;TravelAgent;" />
359 </ExtendedAttributes>
360 </WorkflowProcess>
361 </WorkflowProcesses>
362
363 <ExtendedAttributes>
364     <ExtendedAttribute Name="MadeBy" Value="JaWE" />
365     <ExtendedAttribute Name="Version" Value="1.2" />
366 </ExtendedAttributes>
367 </Package>

```

O pacote *travel* tem três aplicações *HotelApp*, *CarApp*, e *AirApp* (linhas 17-60) e um processo de *workflow*, chamado *travelWorkflow* (cuja definição começa na linha 65). *travelWorkflow* recebe o parâmetro *itinerary* (linhas 72-78) e tem três campos de

dados para os resultados: *airResult*, *hotelResult*, e *carResult* (linhas 79-95). Também há um participante chamado *TRavelAgent* (linhas 96-100).

A primeira atividade no processo é uma atividade manual chamada *GetItinerary* (linhas 107-134), designada para o participante *TRavelAgency*. O agente faz a primeira análise e ativa de modo paralelo as atividades *SubmitAir* (linhas 138-159), *SubmitHotel* (linhas 163-184) e *SubmitCar* (linhas 188-209), as quais chamam as aplicações *AirApp*, *HotelApp* e *CarApp*, passando o itinerário como parâmetro e recebendo o resultado, que vai para os campos de dados já definidos. A atividade *route* (linhas 254-278) sincroniza os resultados das três atividades e divide o fluxo em dois através de *ou exclusivo* para as atividades *MailItinerary* (linhas 213-229) ou *CallCustomerForMods* (linhas 233-249). A escolha de qual direção tomar é determinada pela avaliação de uma condição: a transição *MailItinerary* (linhas 331-339) é tomada somente se *airResult="ok"*, *carResult="ok"* e *hotelResult="ok"*. Caso contrário, a transição *CallCustomerForMods* (linhas 341-347) acontece.